

65250-20740450

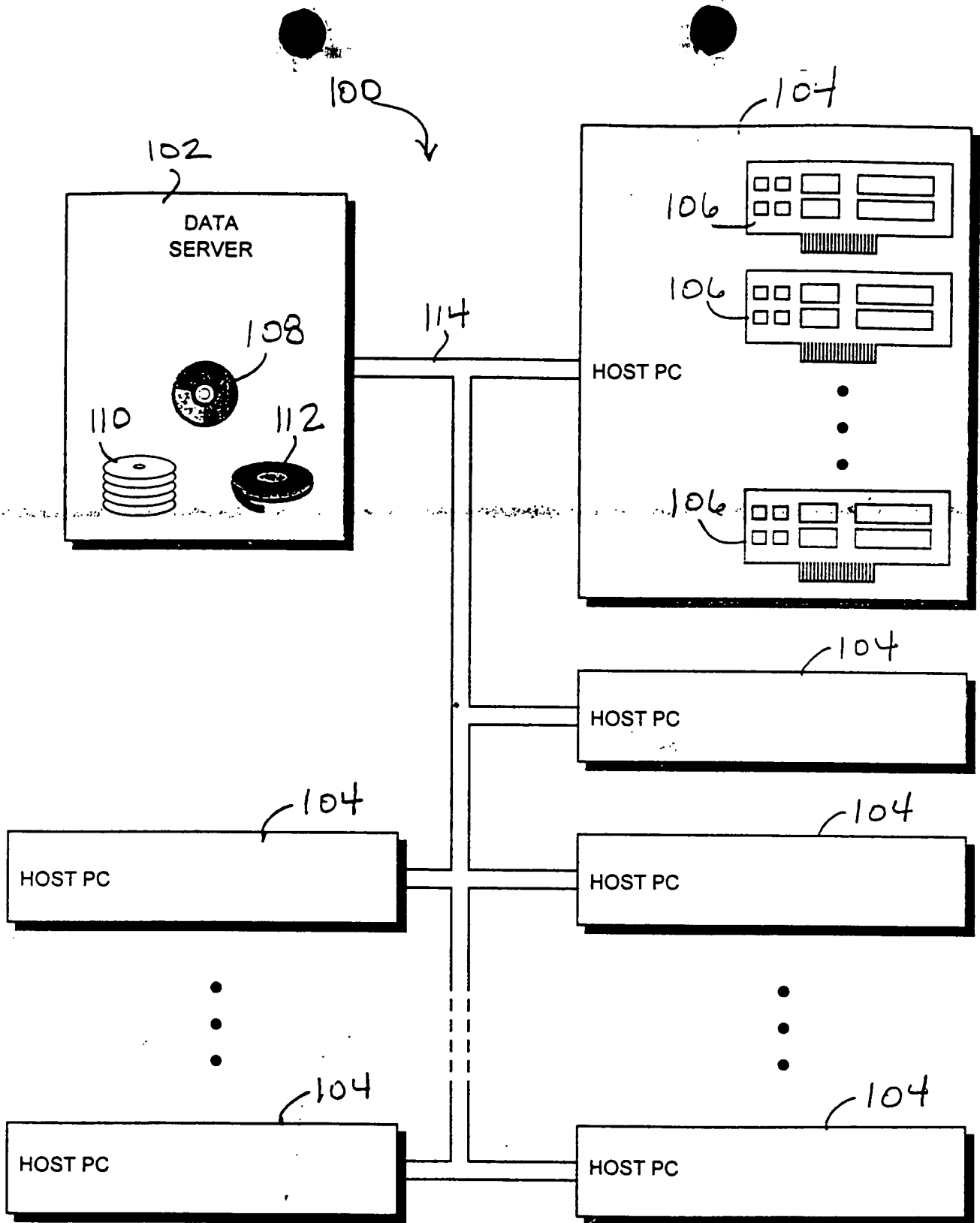


FIG. 1

ALGORITHMIC PROCESSING MODULE

200

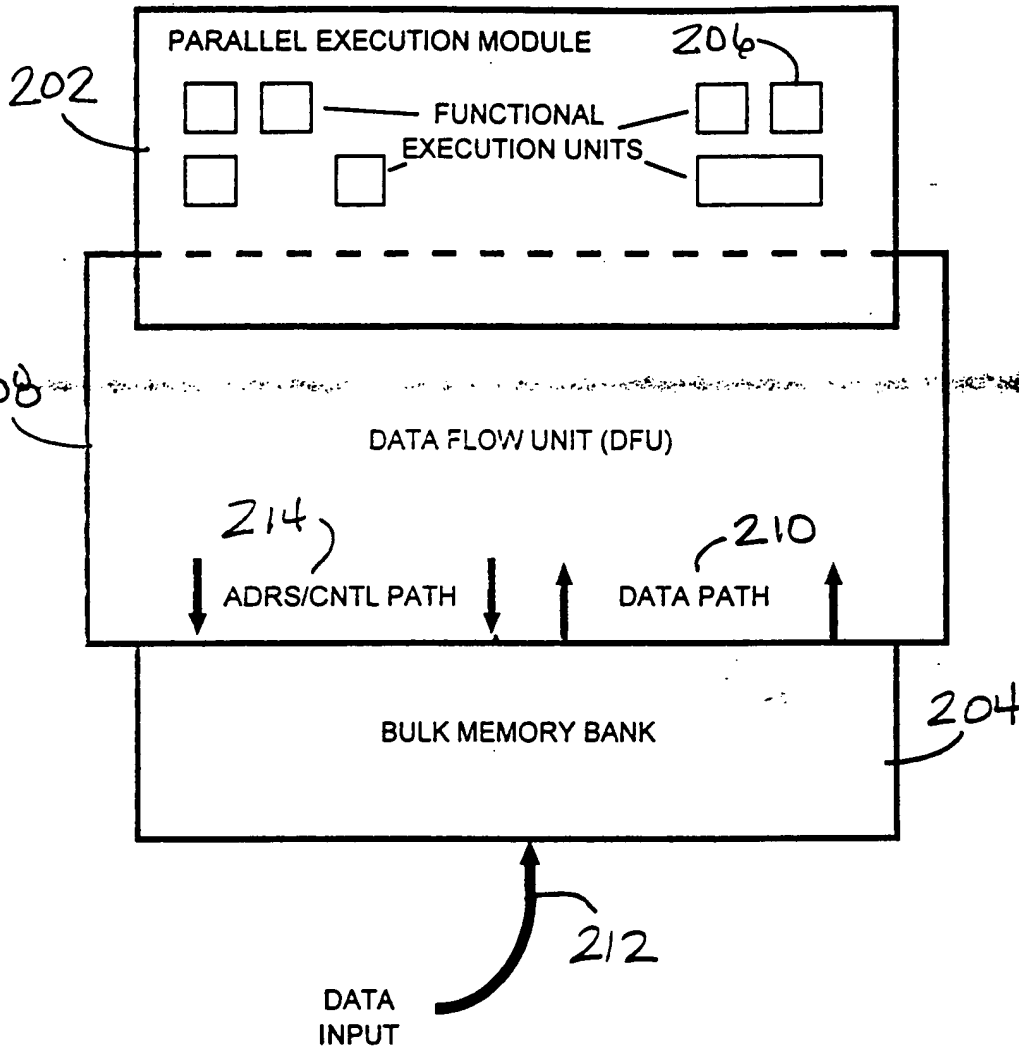


FIG. 2

300
↓

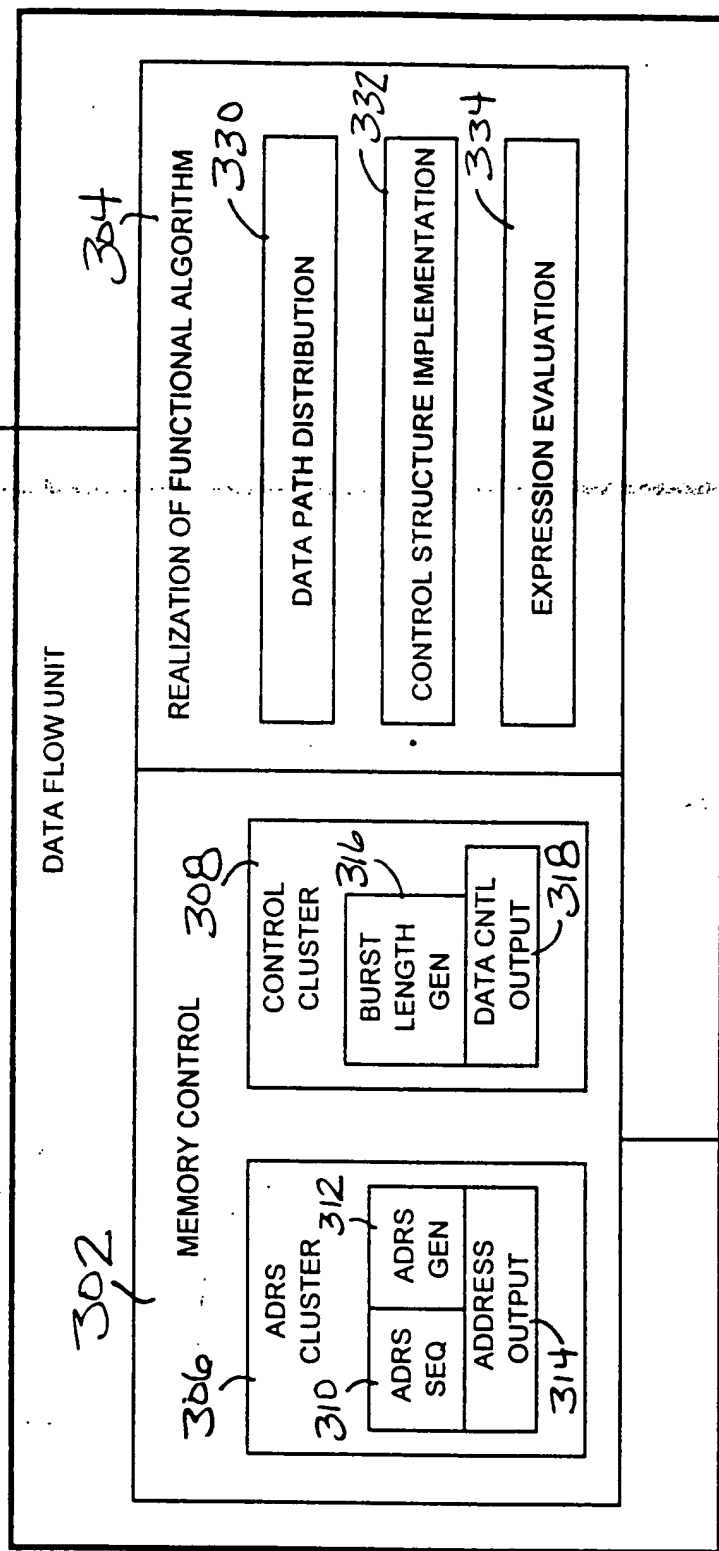


FIG. 3

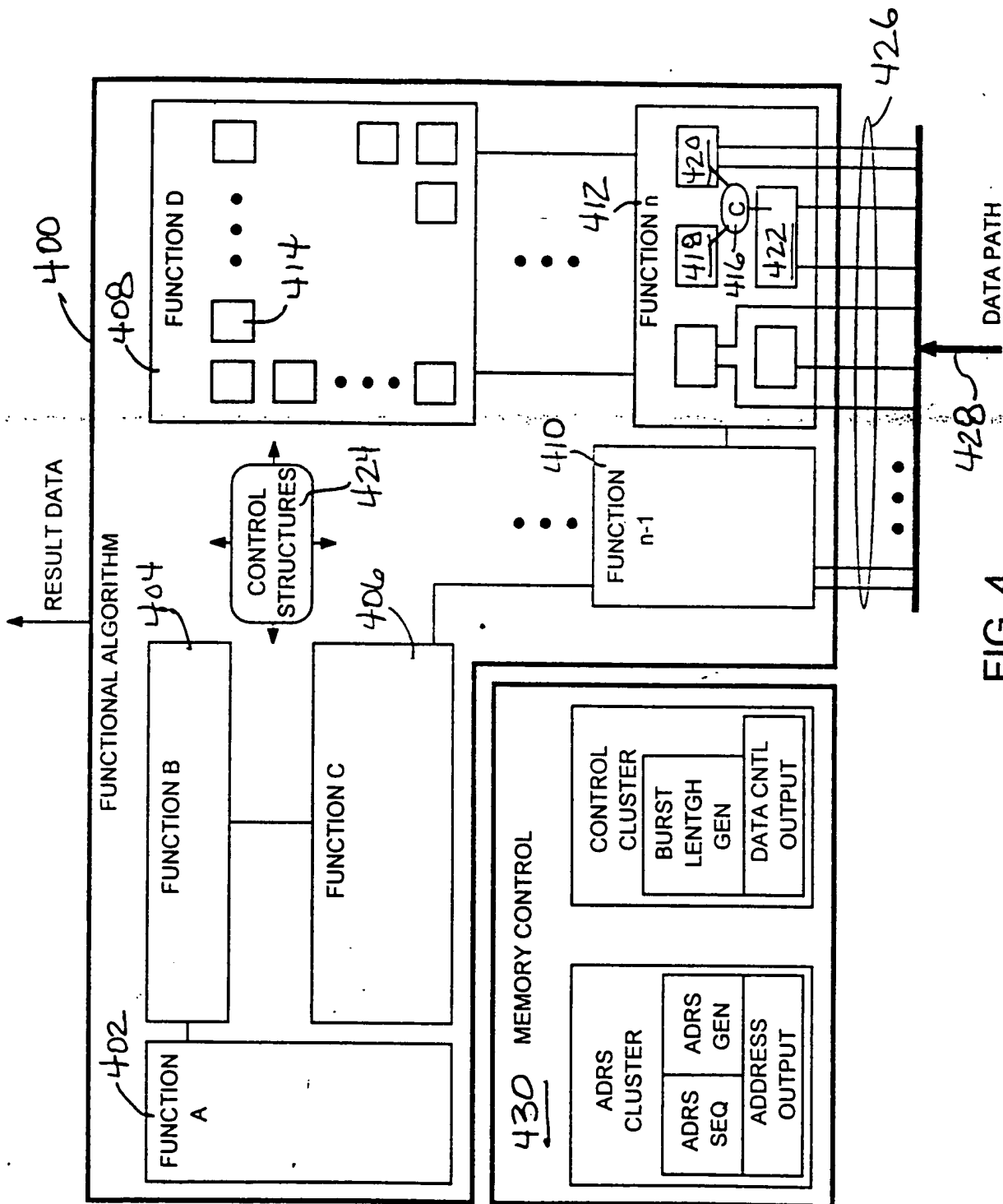


FIG. 4

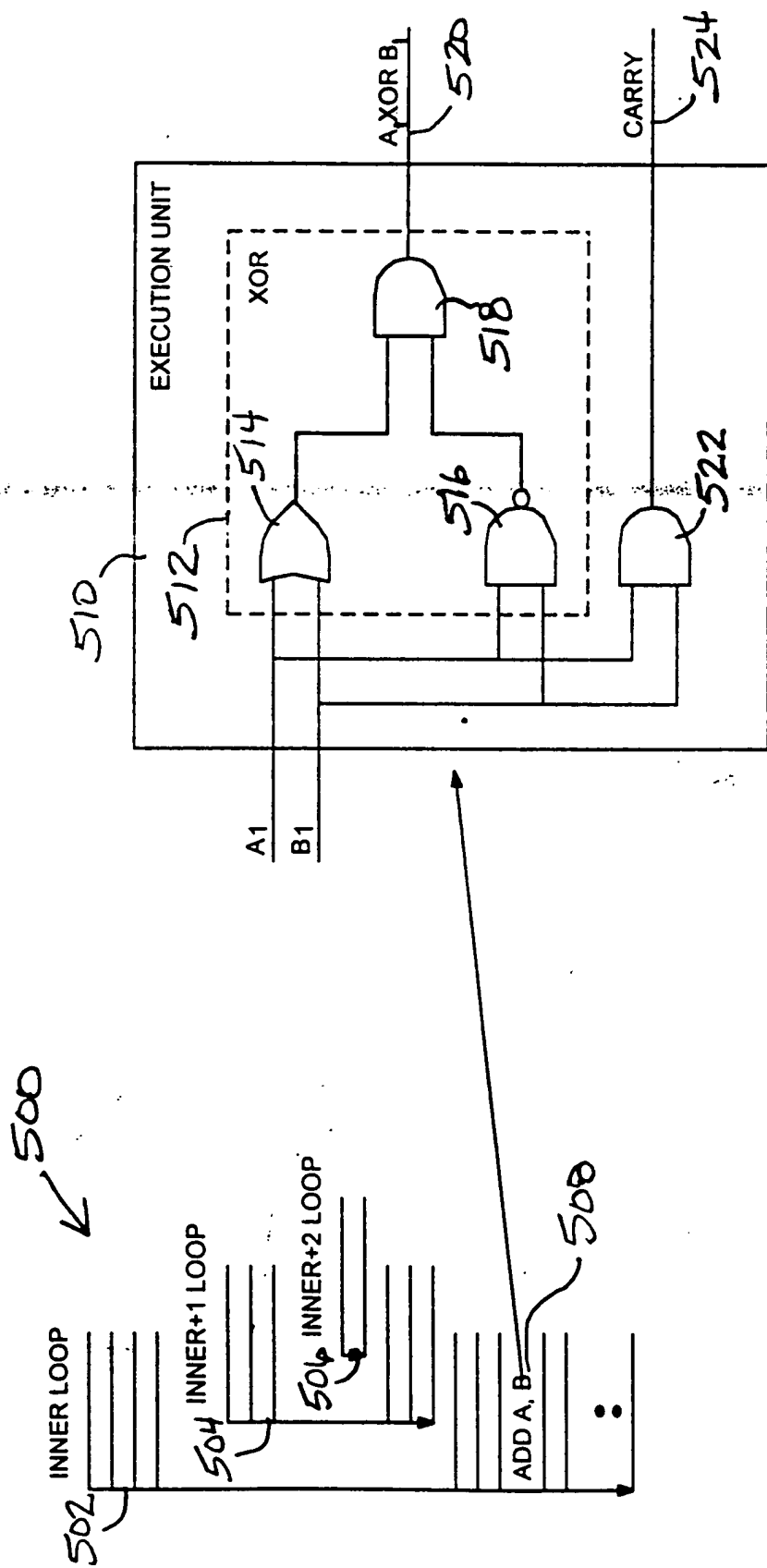


FIG. 5

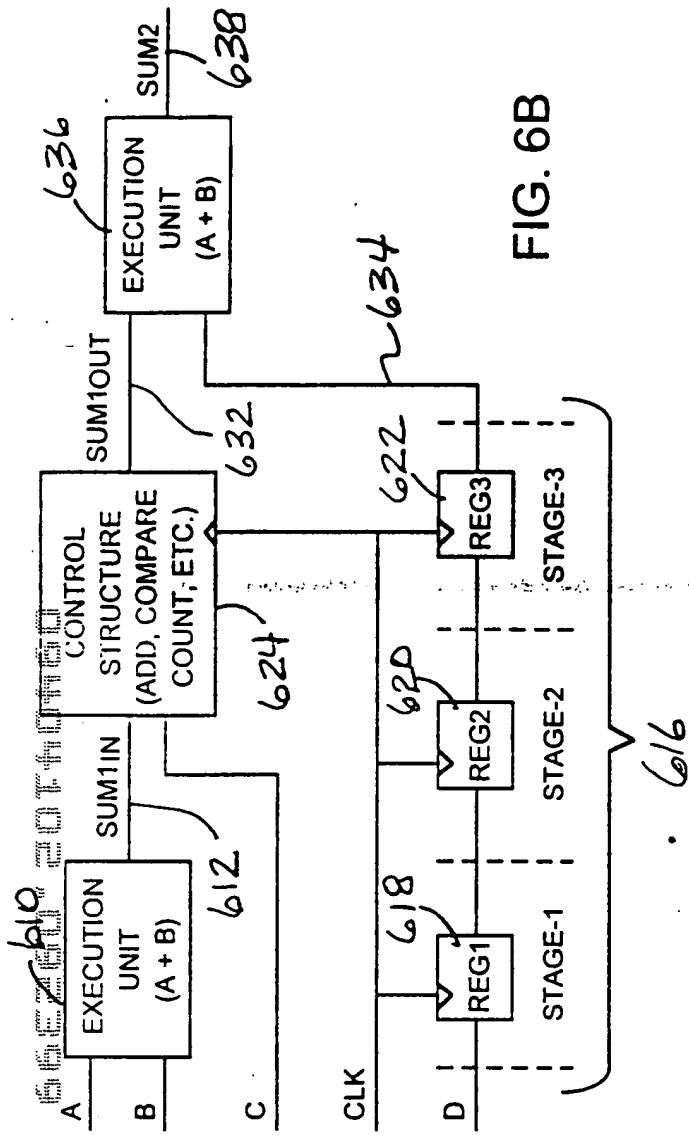


FIG. 6B

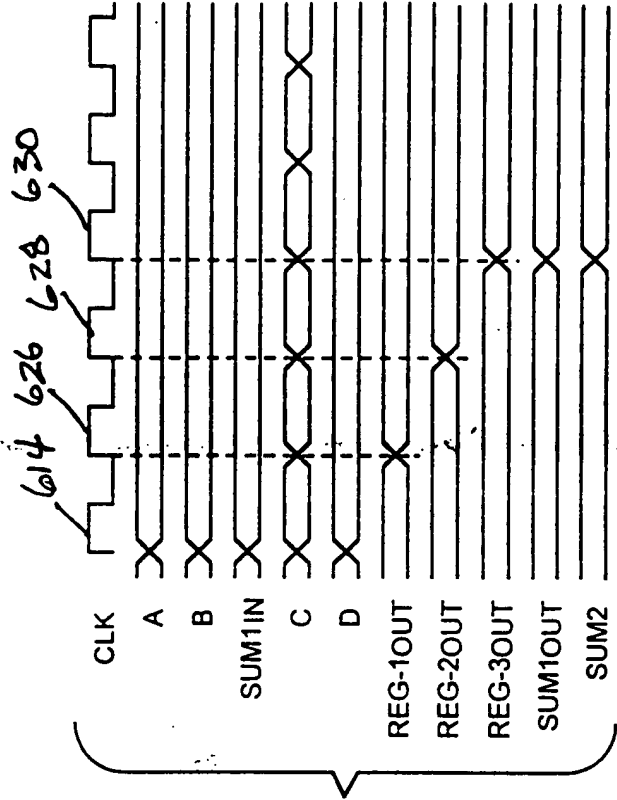


FIG. 6C

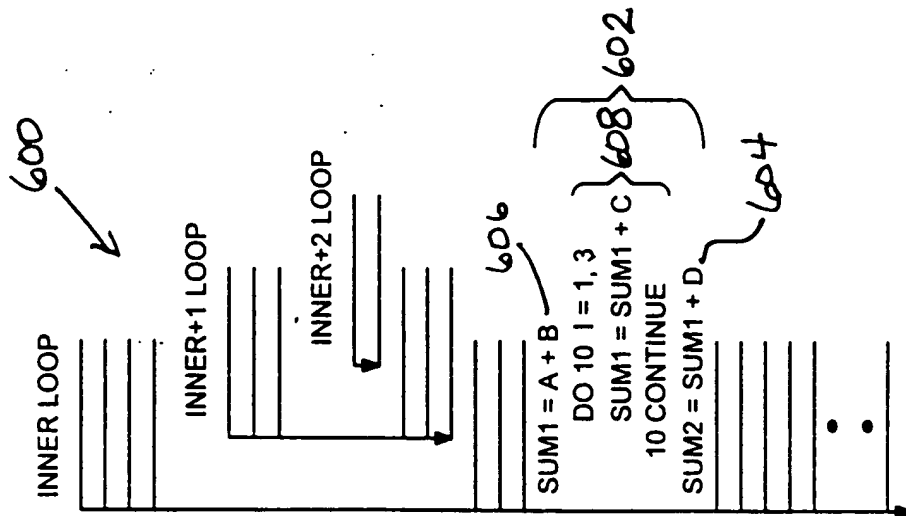


FIG. 6A

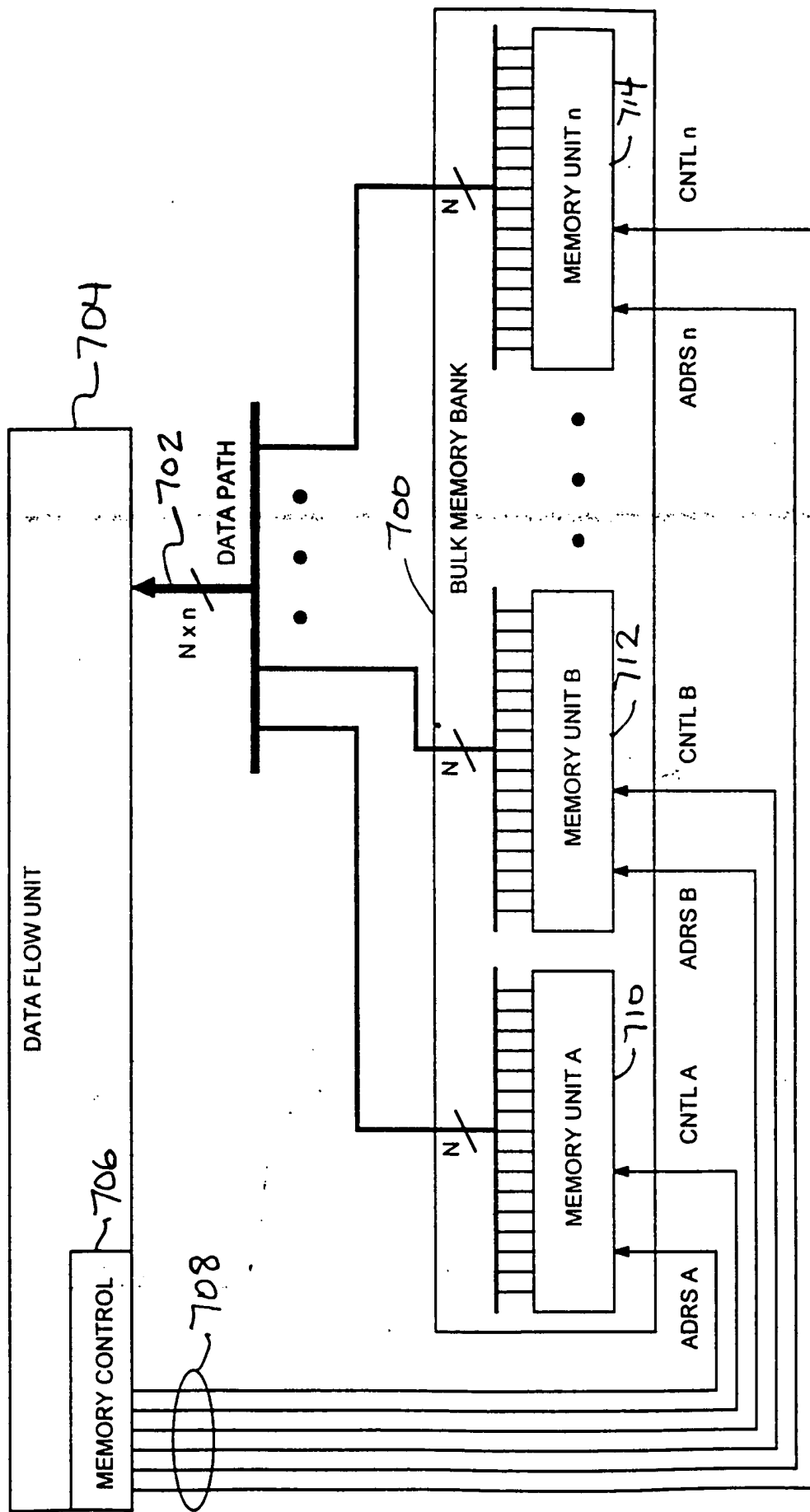


FIG. 7

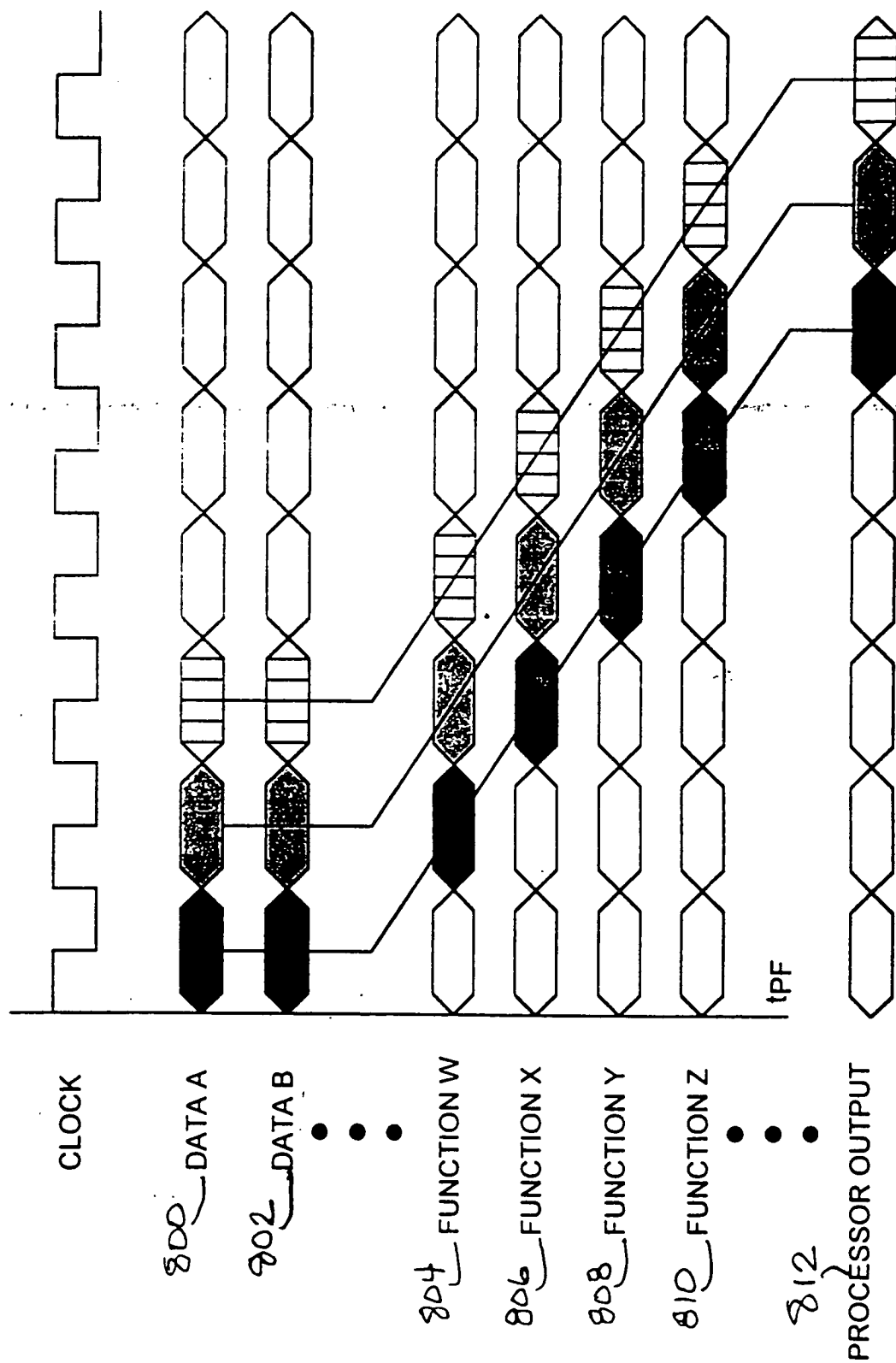


FIG. 8

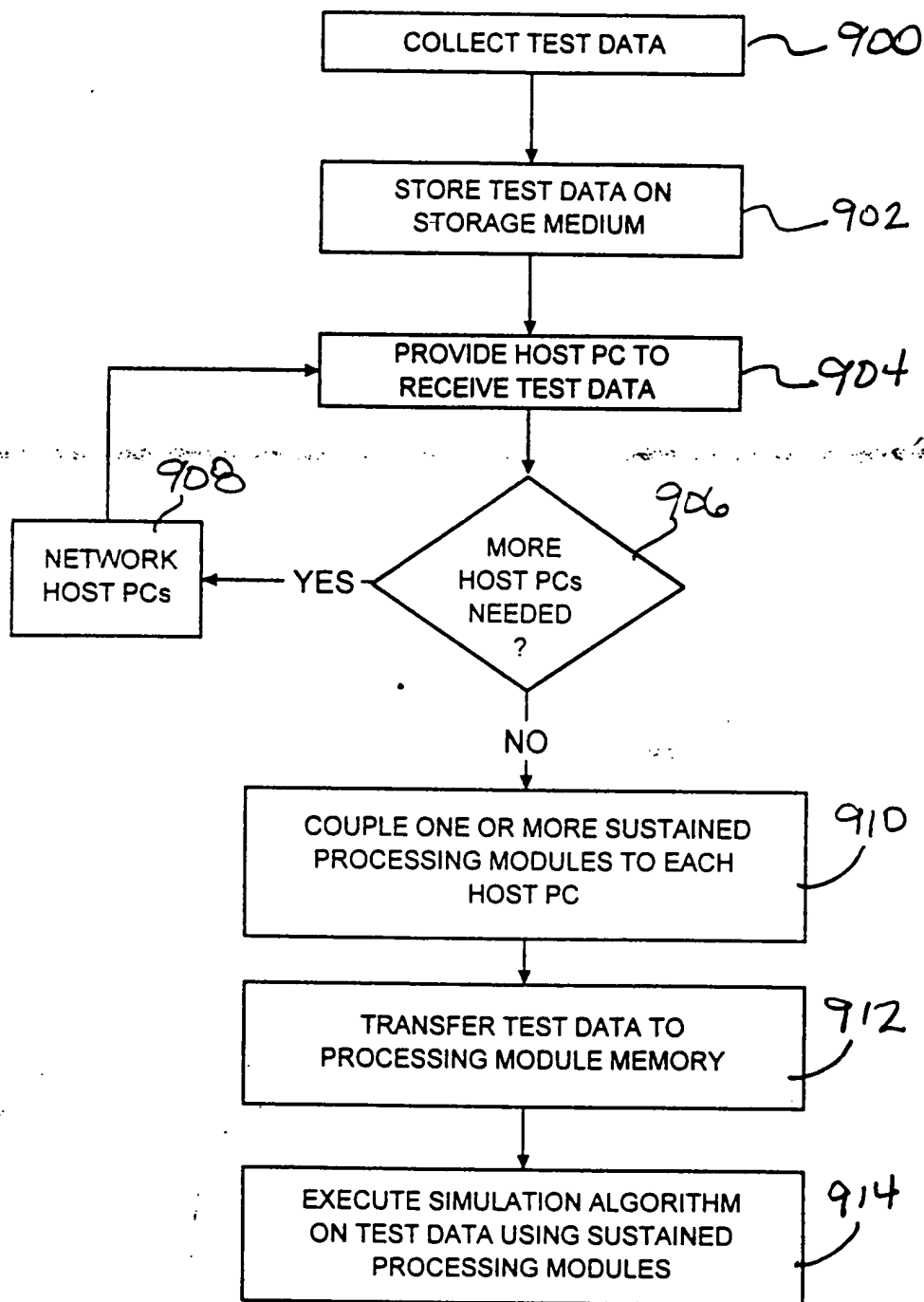


FIG. 9

SUBROUTINE AVERAGE (INDATA, COUNT, MAX, MIN, AVG, RMS)

INTEGER COUNT

REAL INDATA (COUNT), MAX, MIN, AVG, RMS

C

C These are the inputs and outputs to the function

C name	type	mode	length	description
C ====	====	====	=====	=====

C

C INDATA	R	In	COUNT	Array of input values
C COUNT	I	In		Number of values to average
C MAX	R	InOut		Maximum value found
C MIN	R	InOut		Minimum value found
C AVG	R	Out		Average of all values
C RMS	R	Out		Root-Mean-Square of all values

C

REAL SUM, RMSSUM

INTEGER I

C These are internal variables

C Initialization of internal variables

SUM = 0.0

RMSSUM = 0.0

C Main Loop proper

DO 100 I = 1, COUNT

IF (INDATA(I) .LT. MIN) THEN

MIN = INDATA(I)

ENDIF

IF (INDATA(I) .GT. MAX) THEN

MAX = INDATA(I)

ENDIF

SUM = SUM + INDATA(I)

RMSSUM = RMSSUM + INDATA(I) * INDATA(I)

100 CONTINUE

C End of main loop

C Calculate average, RMS

AVG = SUM / COUNT

RMSSUM = SQRT(RMSSUM/COUNT)

END

FIG. 10

66E250*20T40460

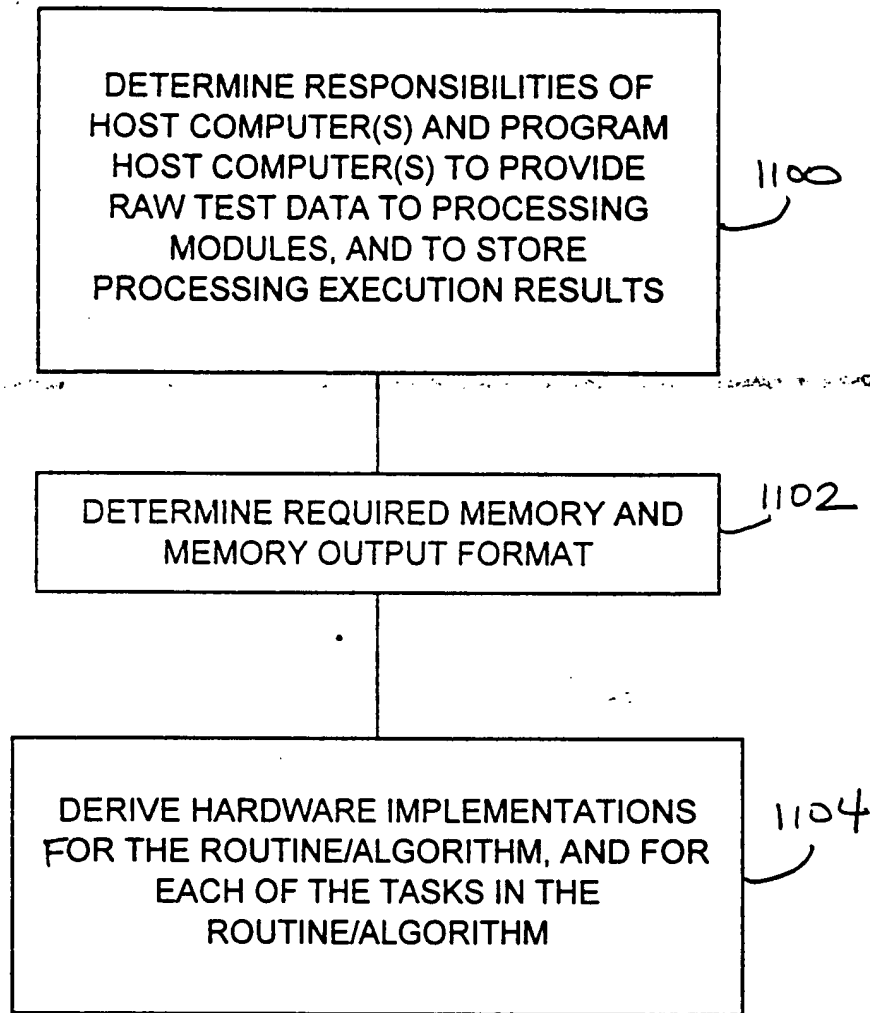


FIG. 11

```

IF (INDATA(I) .LT. MIN) THEN
  MIN = INDATA(I)
ENDIF

```

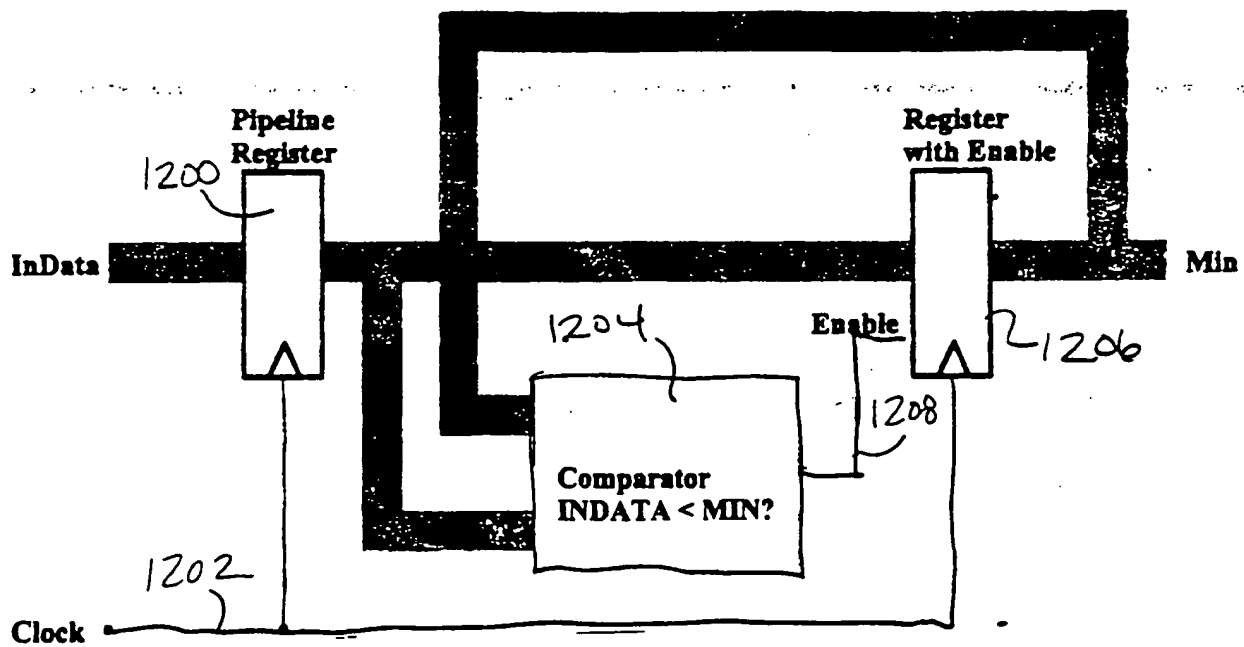


FIG. 12

```

IF (INDATA(I) .LT. MIN) THEN
  MIN = INDATA(I)
ENDIF

```

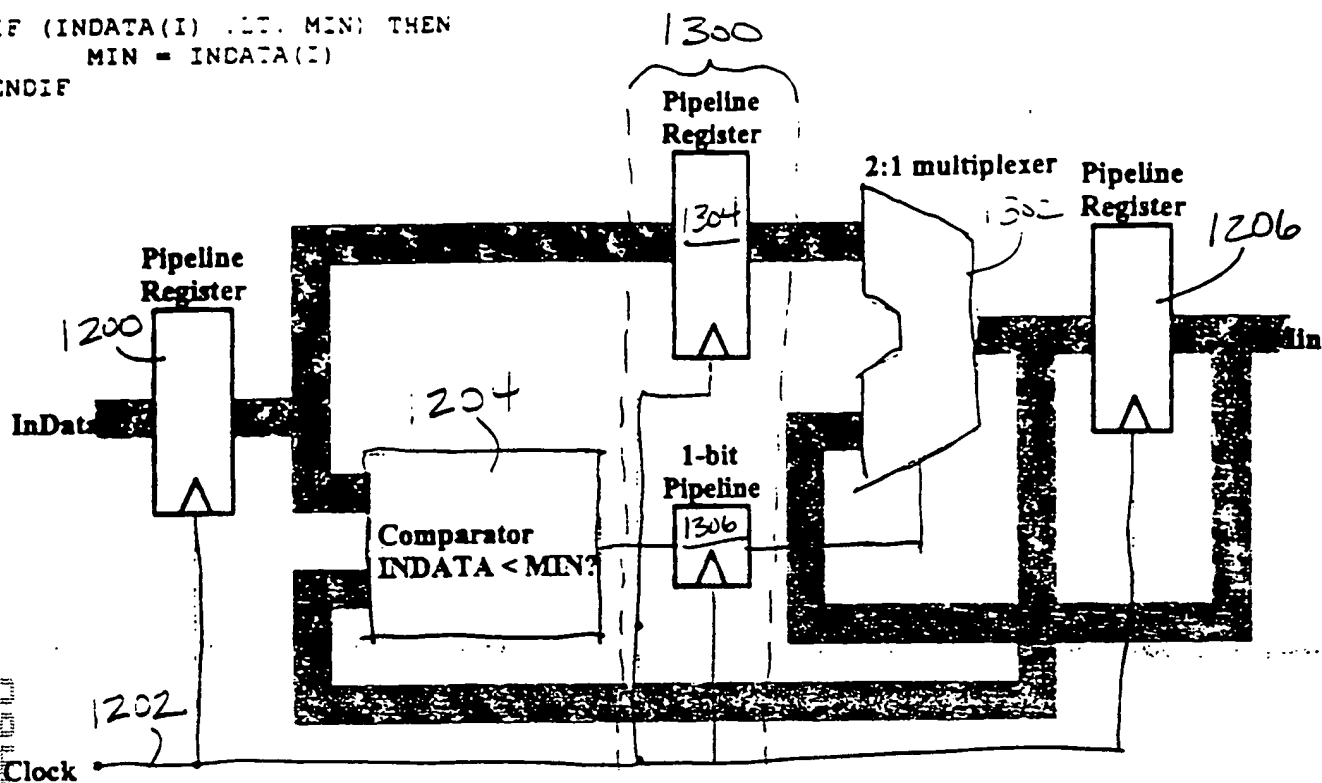


FIG. 13

```

IF (INDATA(I) .LT. MIN) THEN
  MIN = INDATA(I)
ENDIF

```

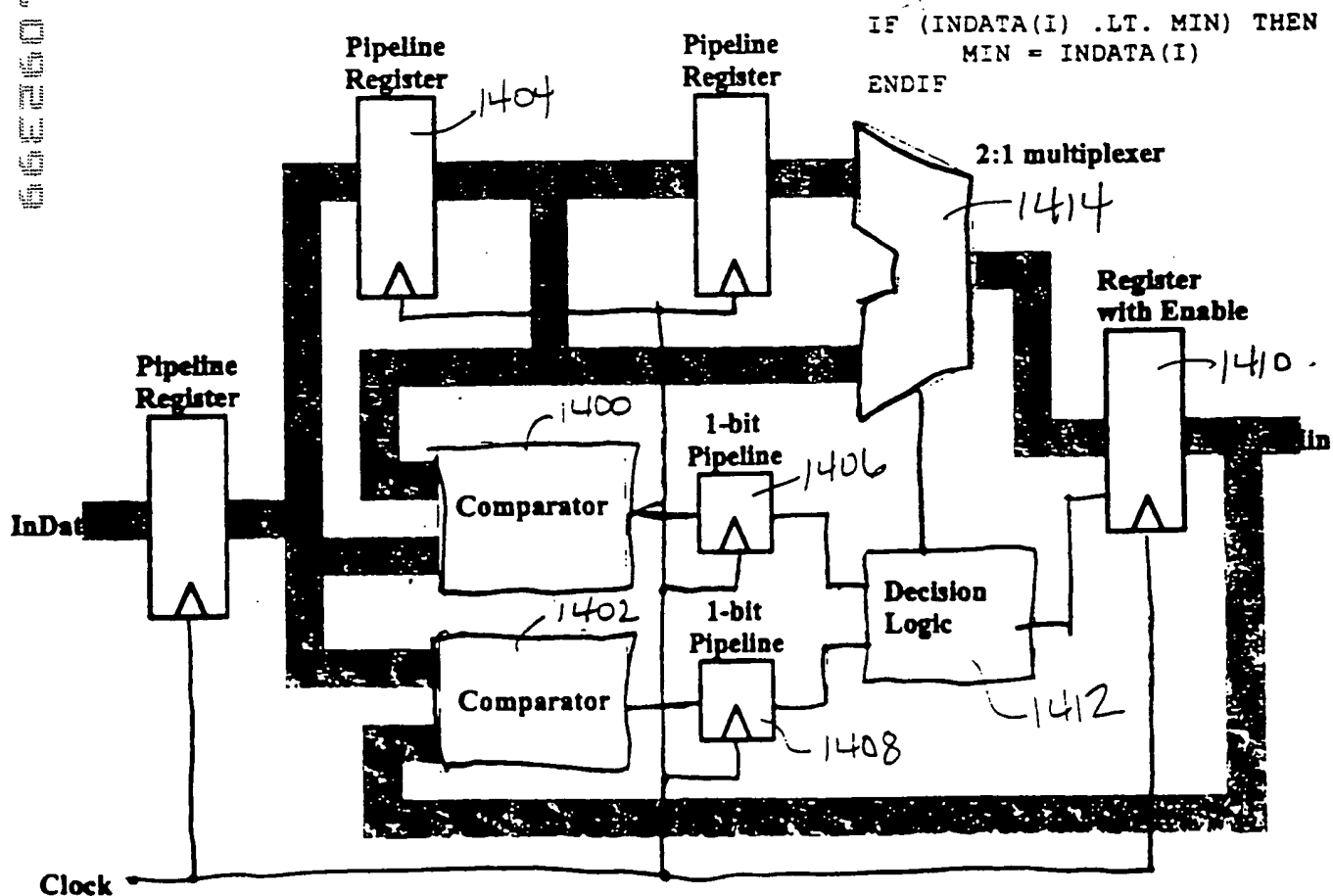


FIG. 14

66E260-20740460

$$\text{SUM} = \text{SUM} + \text{INDATA}(I)$$

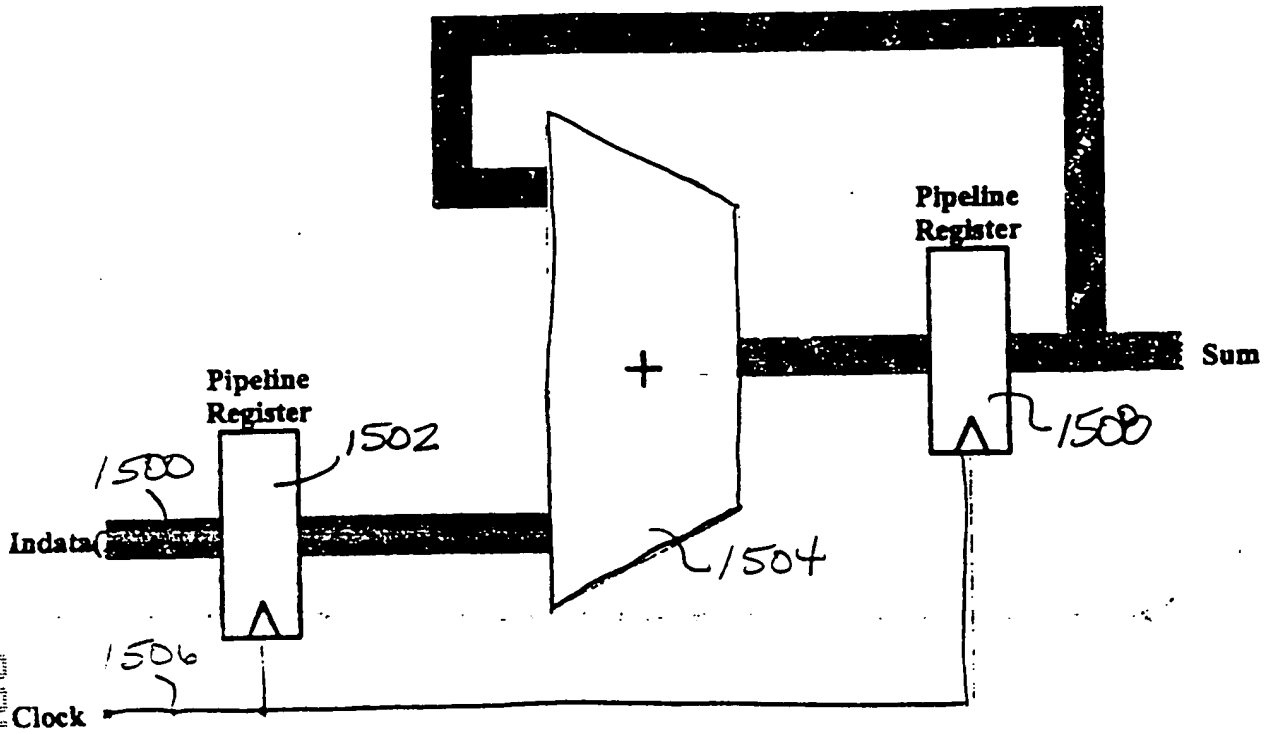


FIG. 15

$$\text{RMSSUM} = \text{RMSSUM} + \text{INDATA}(I) * \text{INDATA}(I)$$

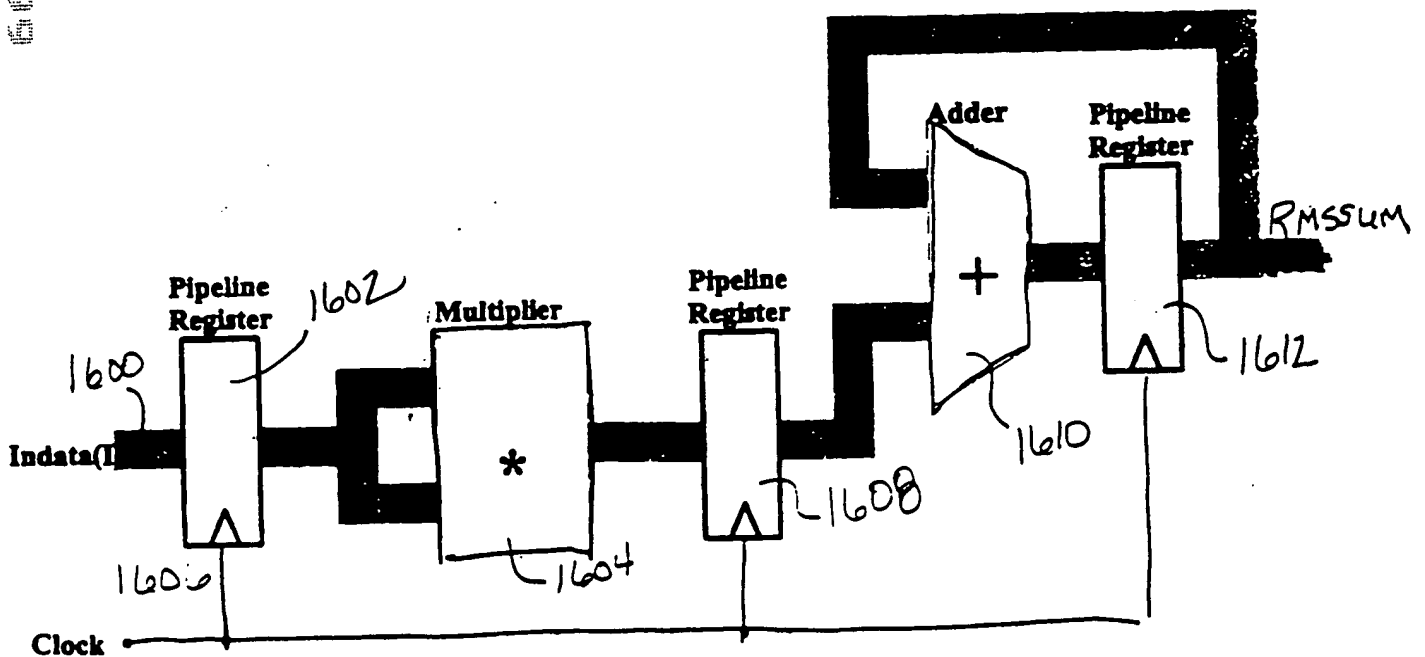


FIG. 16

66E260" 20T40460

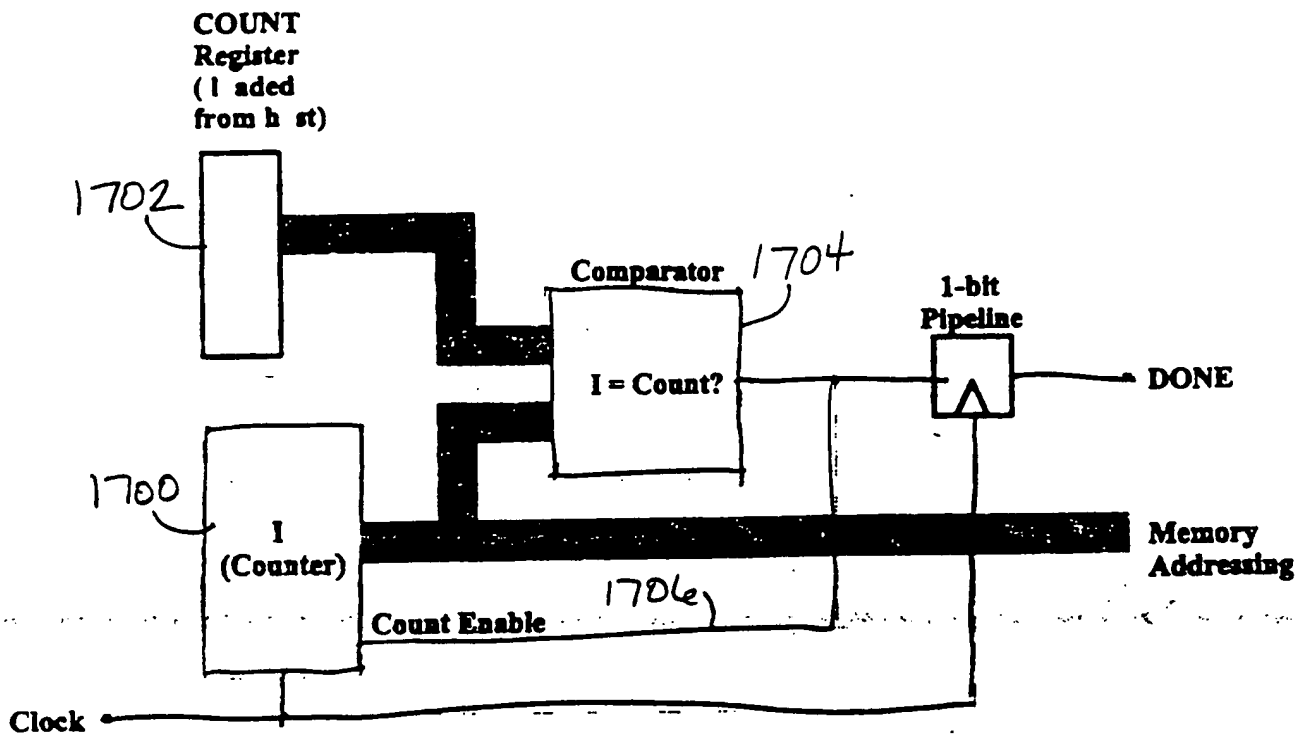


FIG. 17

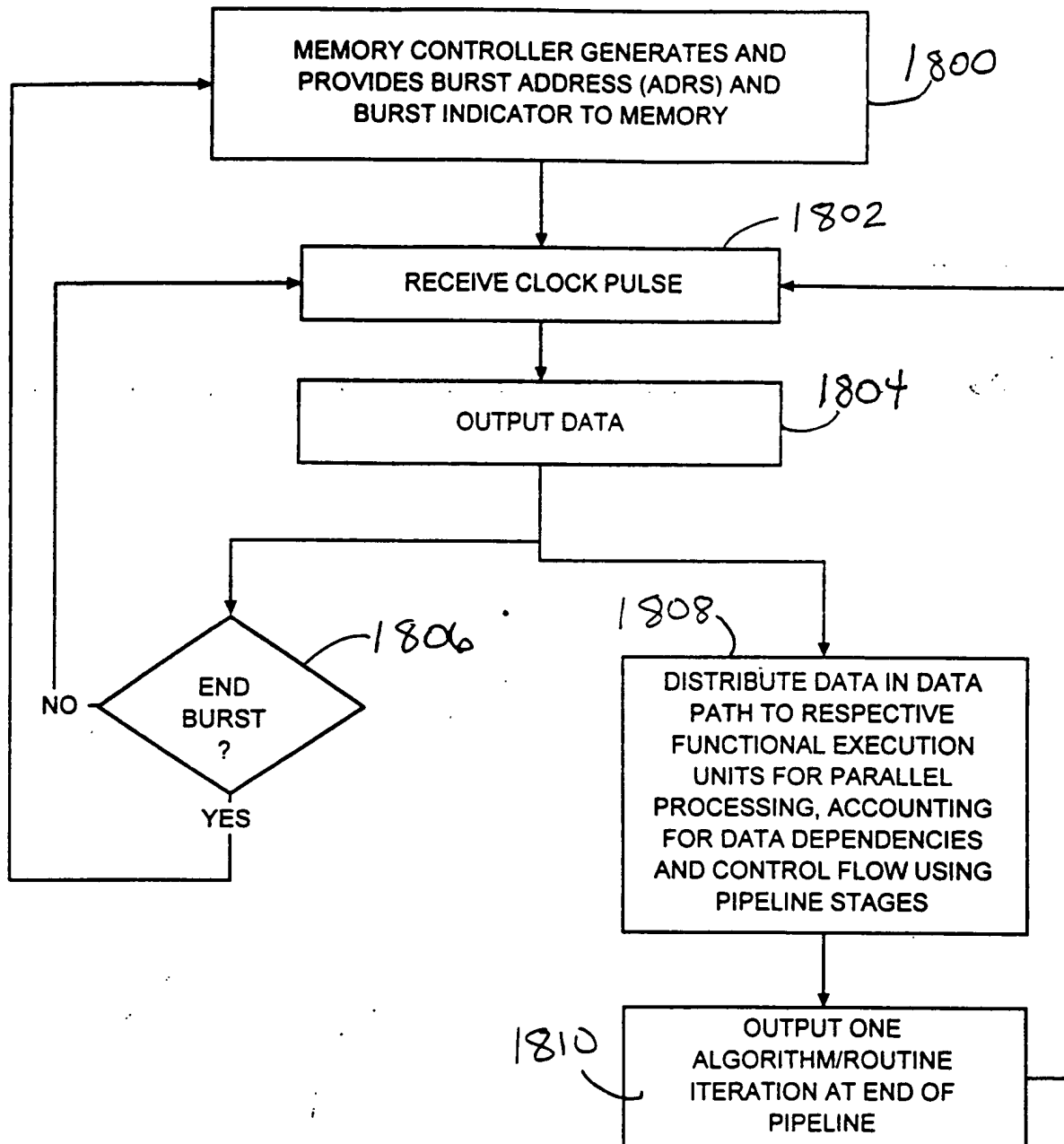


FIG. 18

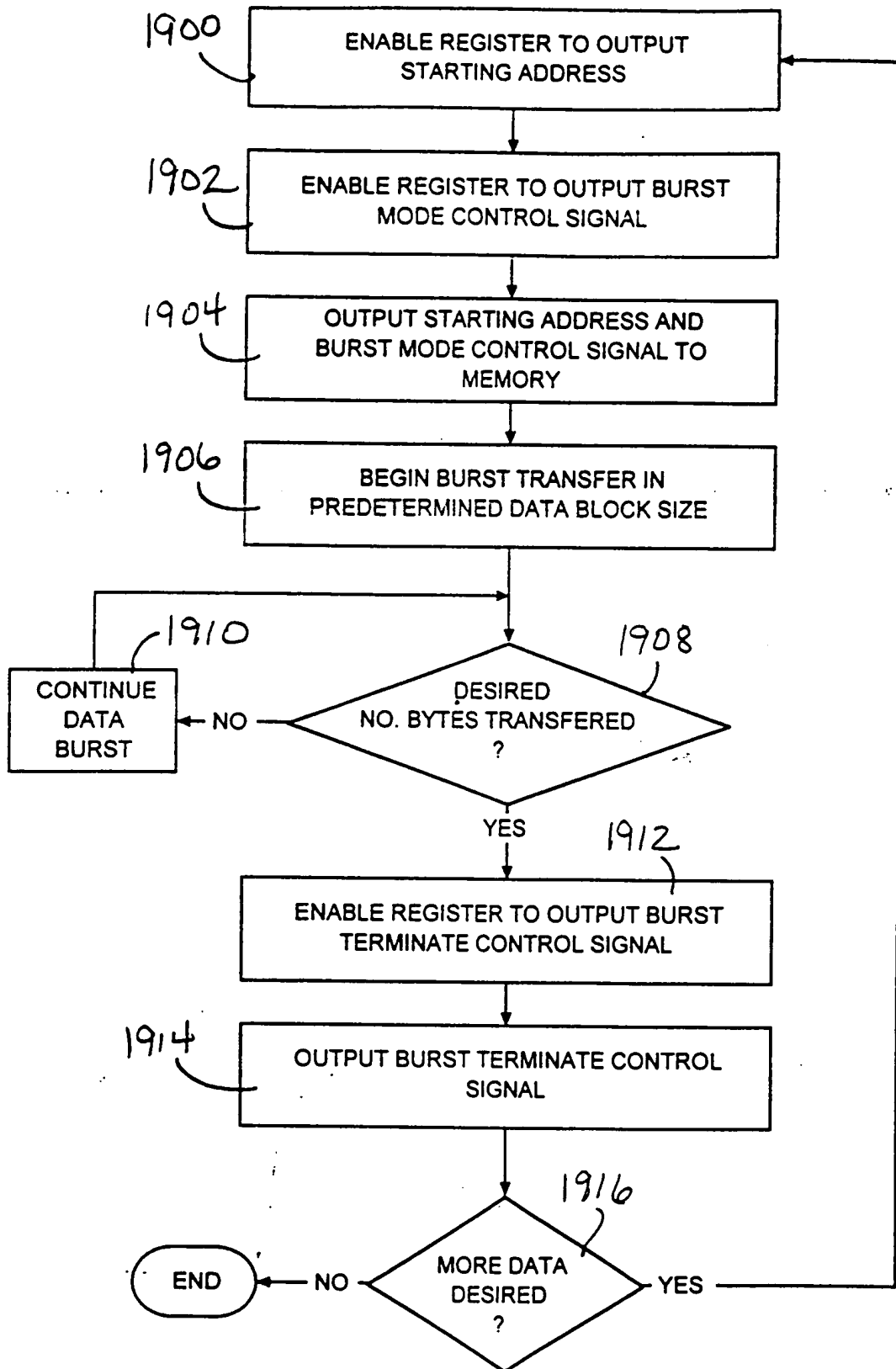


FIG. 19

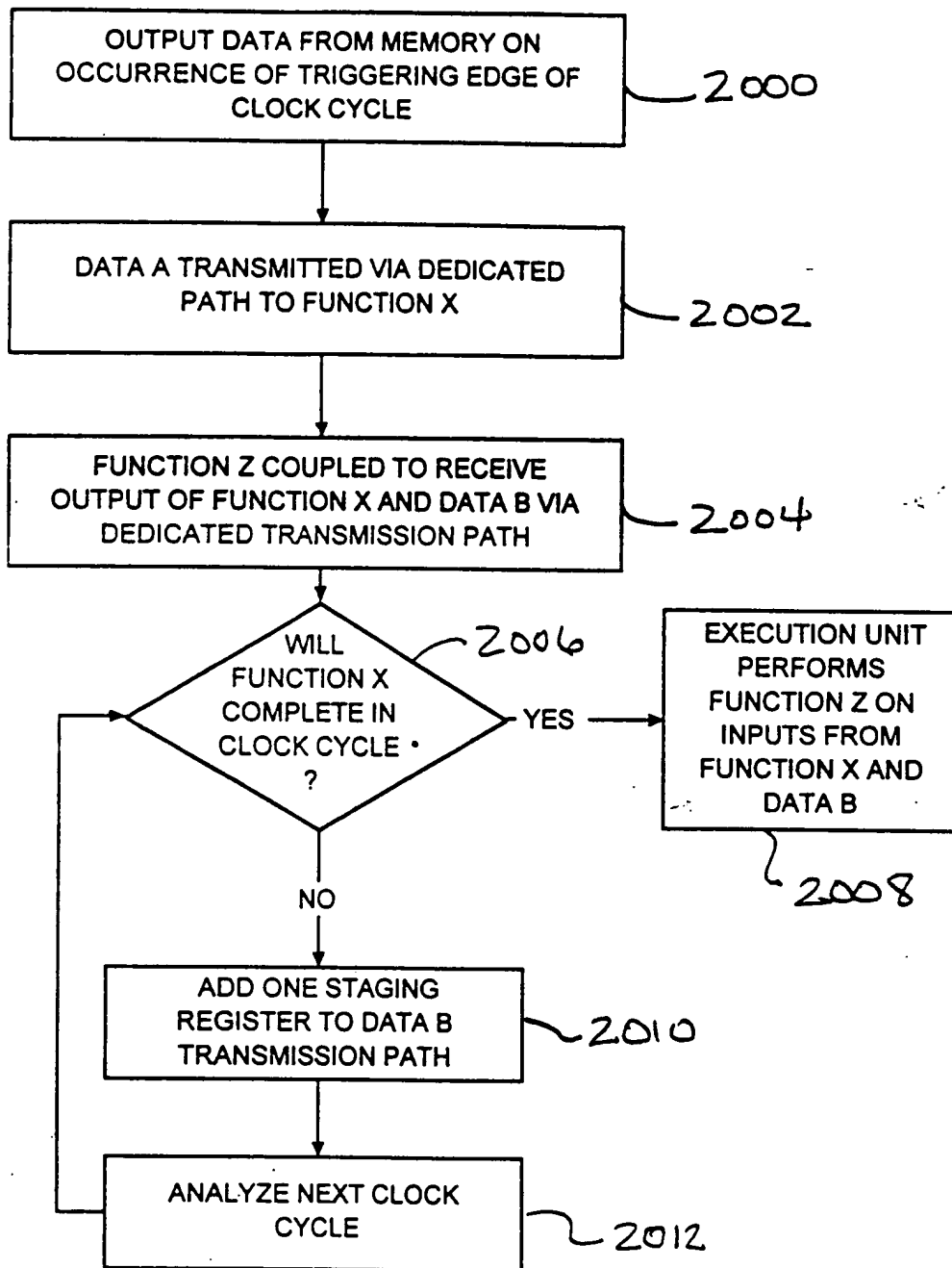


FIG. 20

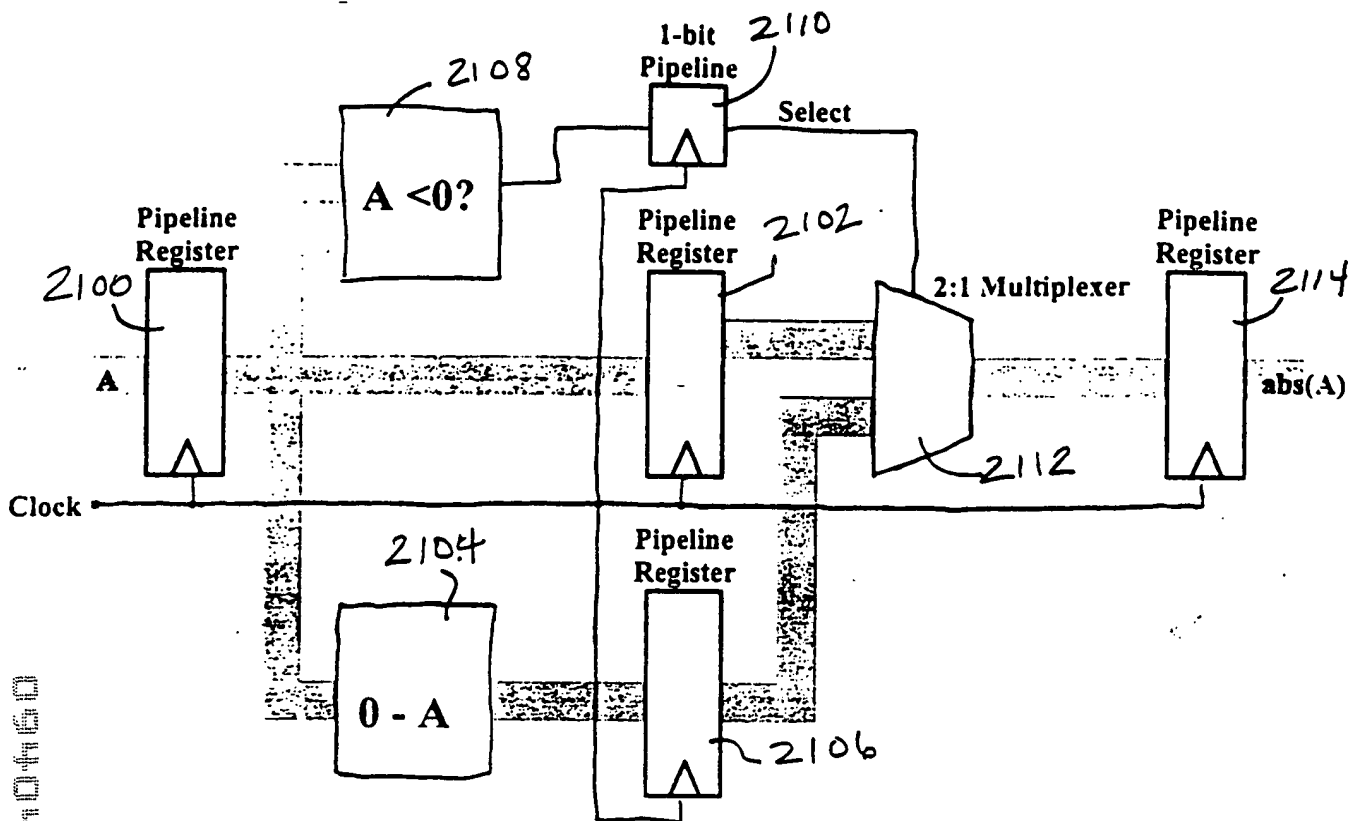


Figure 21. $\text{abs}(a)$ naive hardware implementation - 2 cycles

Note:- A width = N bits

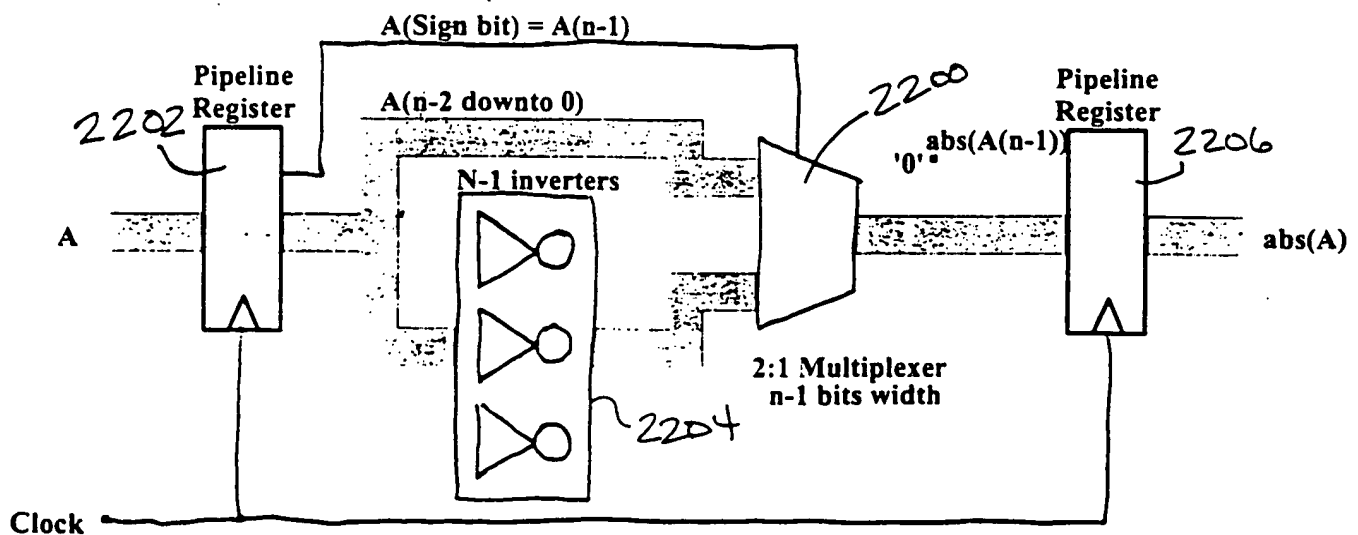


Figure 22. $\text{abs}(a)$ faster implementation - 1 cycle

66E260-20T40460

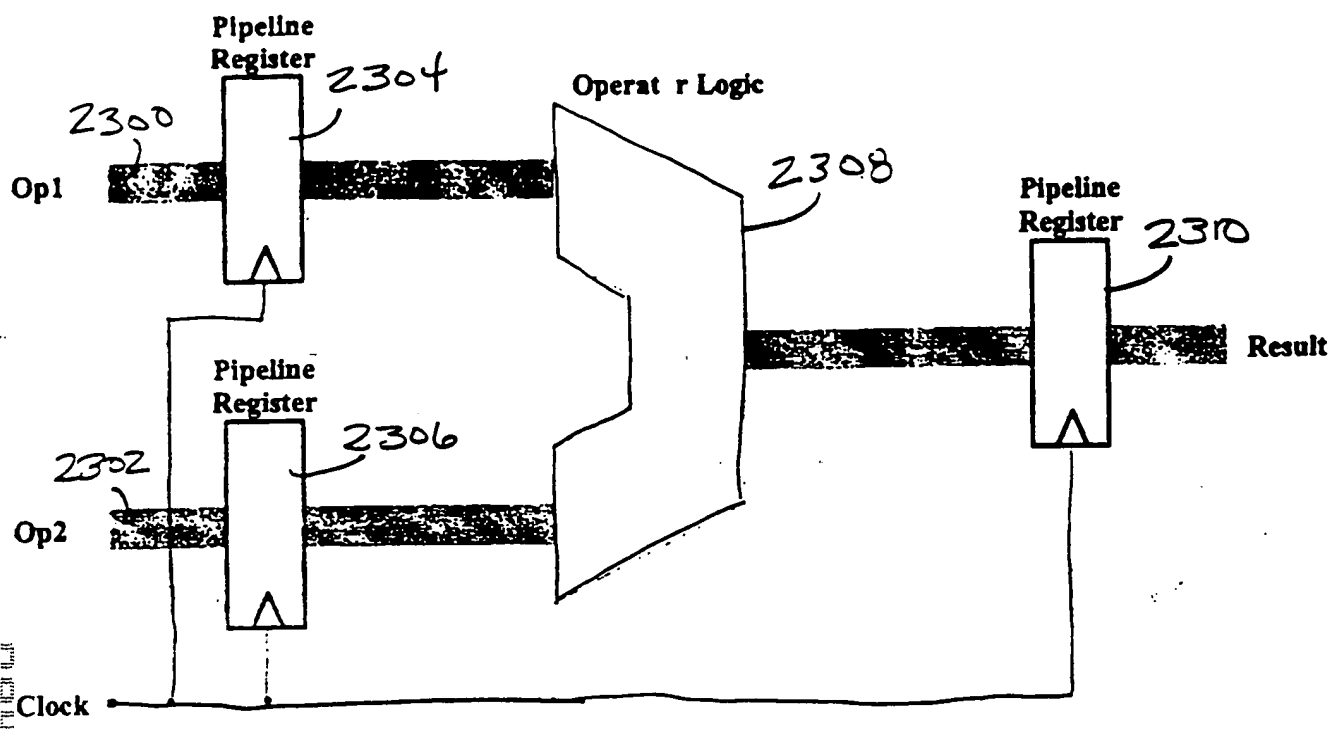


Figure 23 Binary operator

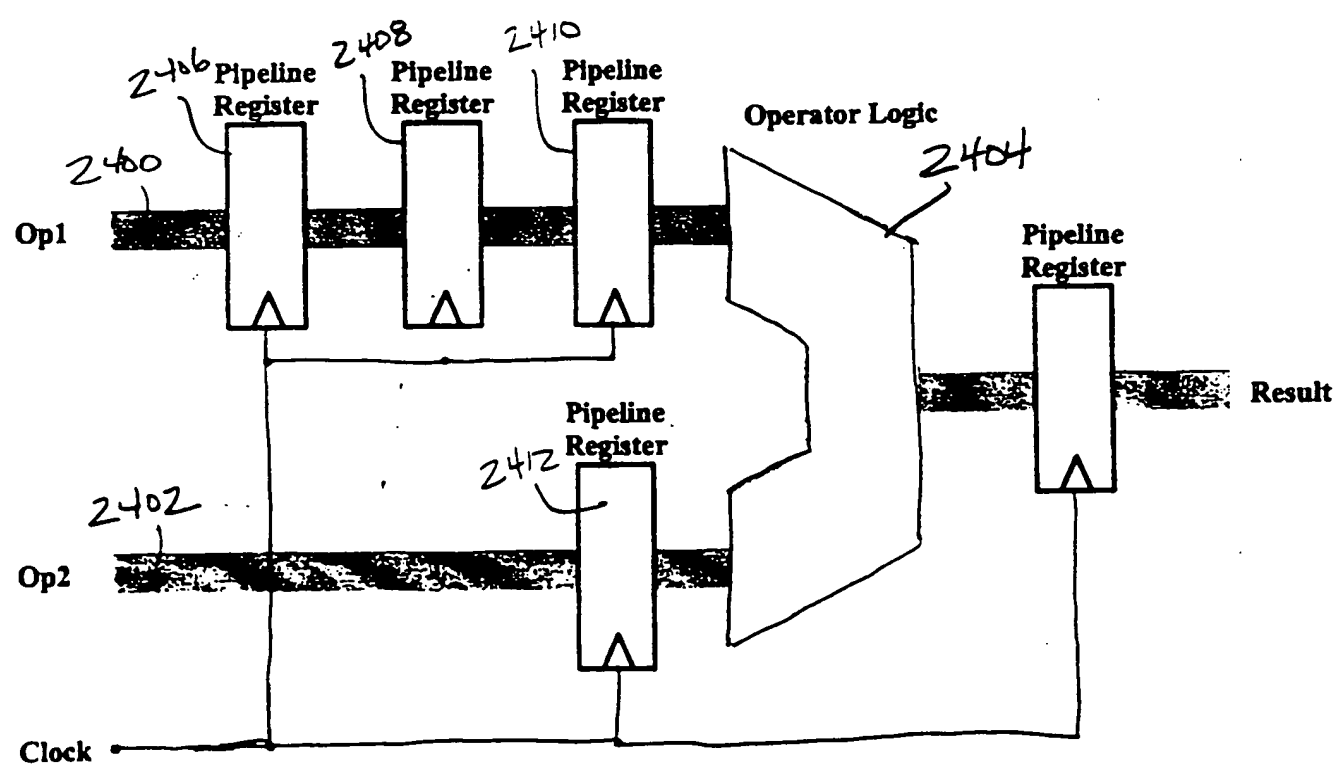


Figure 24 Binary operator with one early operand

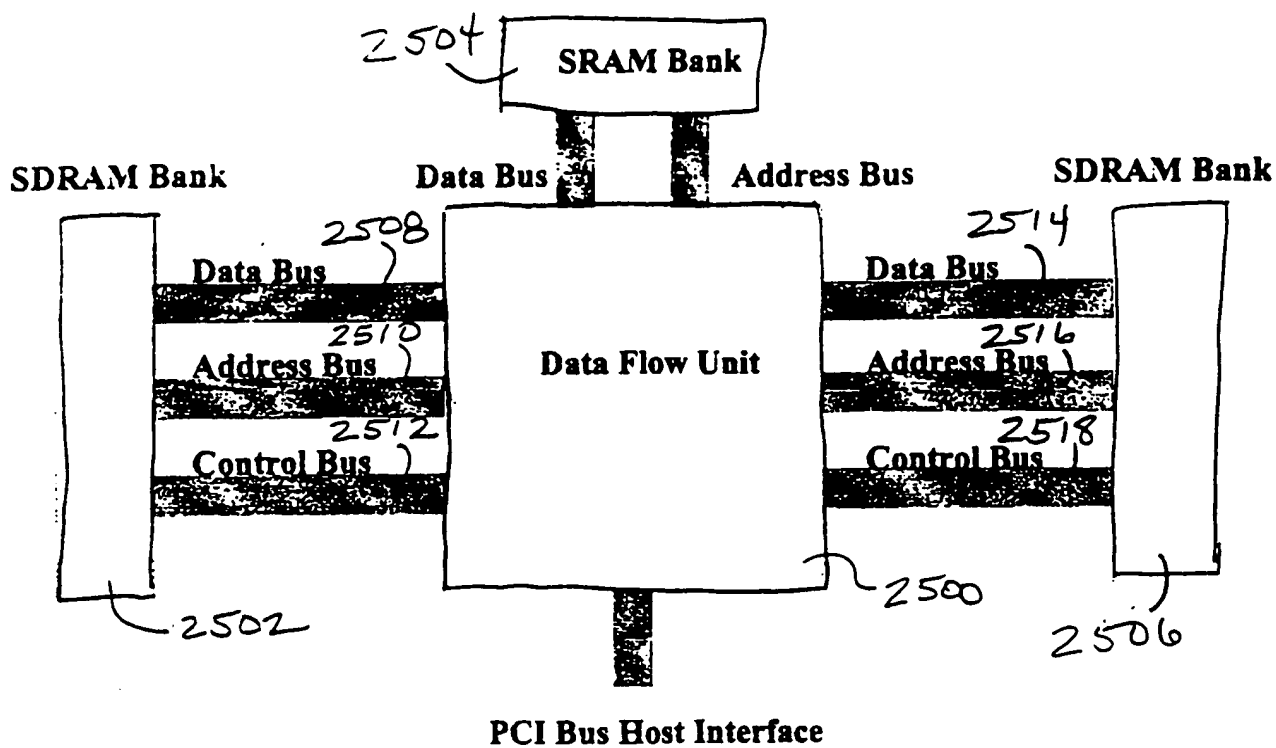


Figure 25 Typical system block diagram

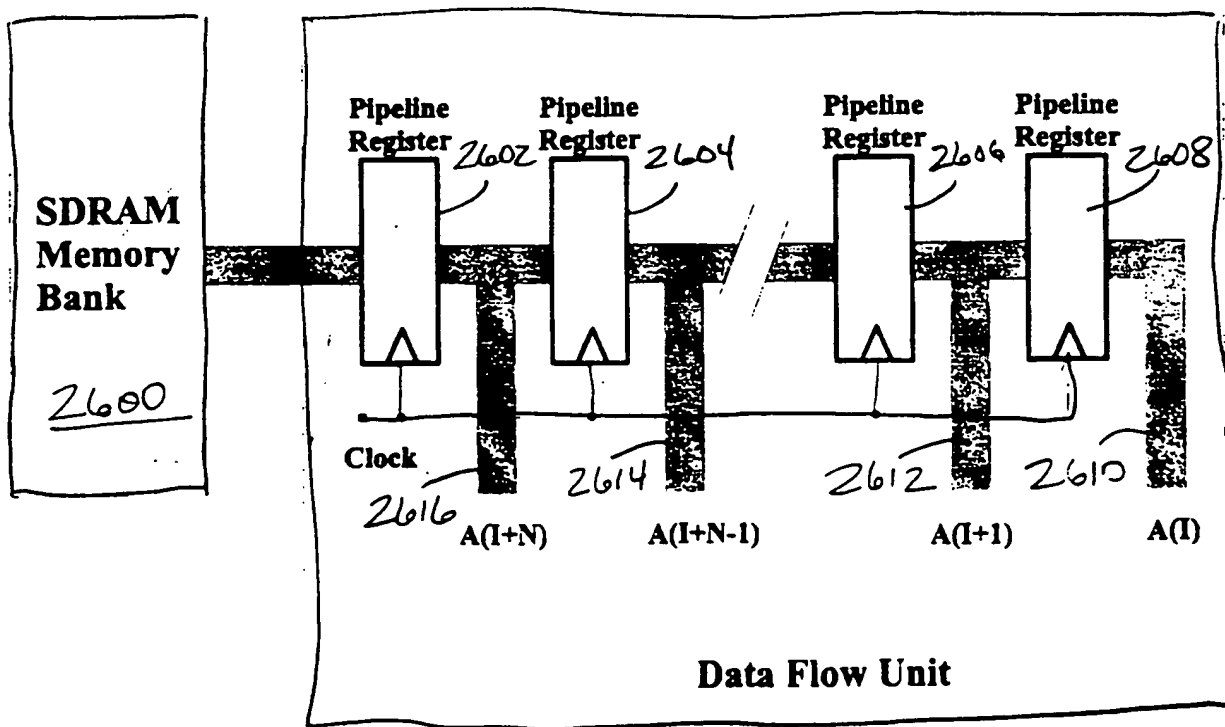


Figure 26 Pipelining to reduce memory accesses.

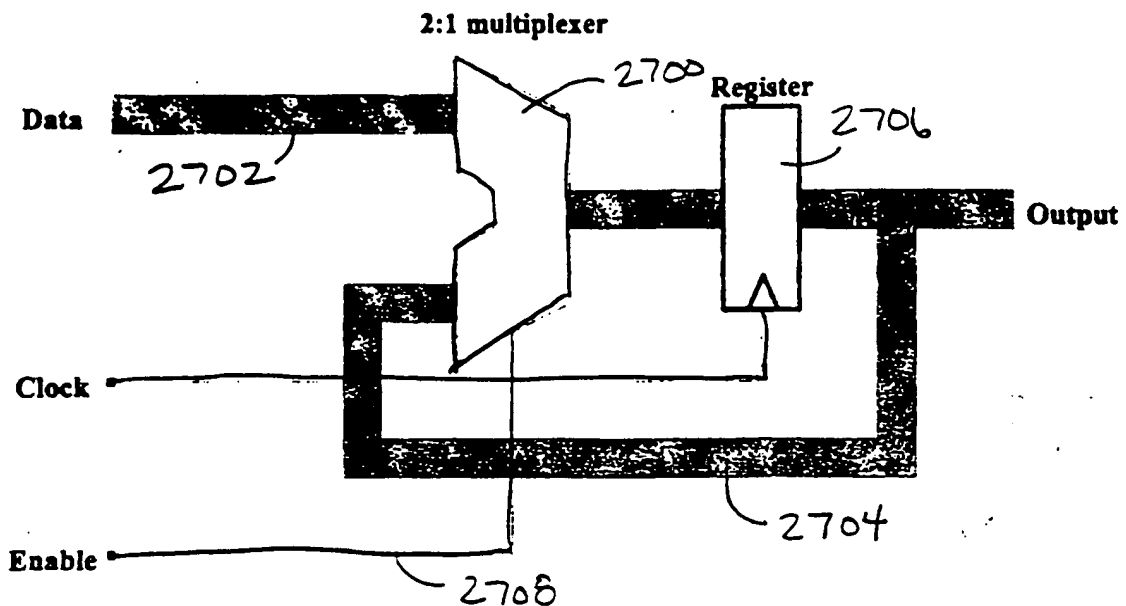


Figure 27 Implementation of Register with Enable

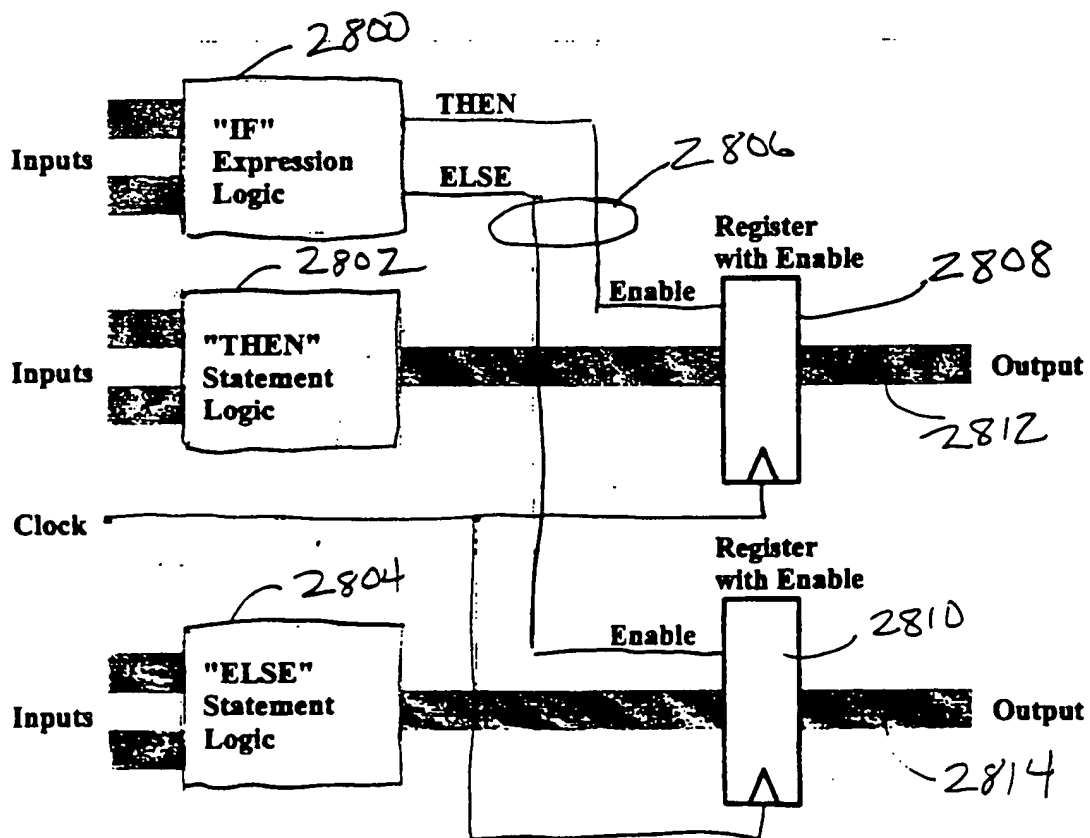


Figure 28 Hardware implementation of conditional statement.

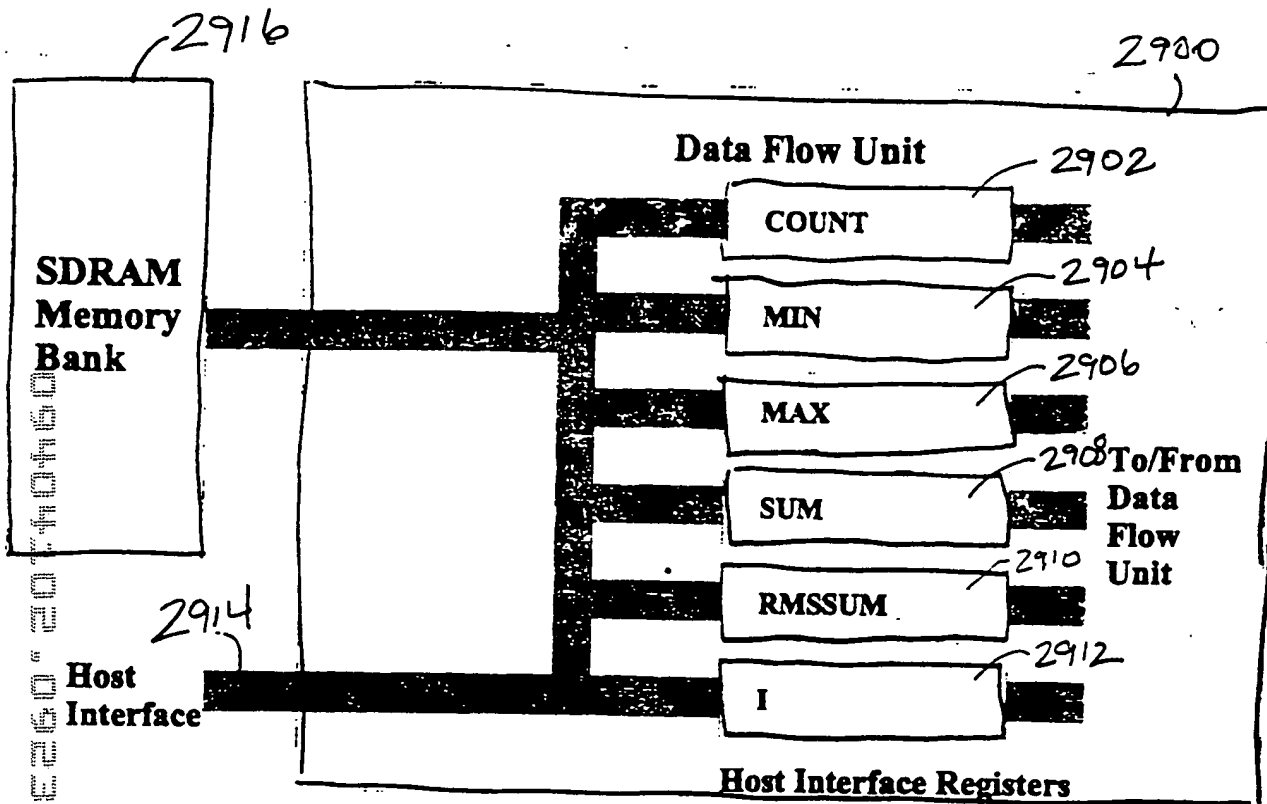


Figure 29 Example host interface/programming model

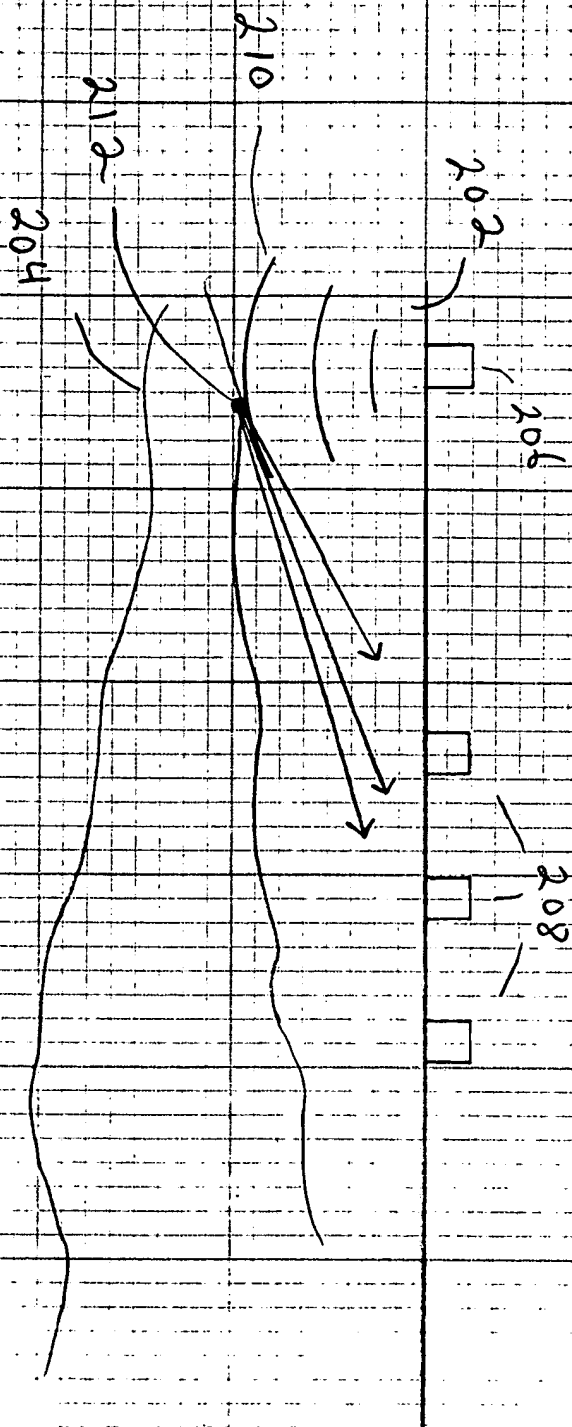


Fig. 30

09404102, 092399

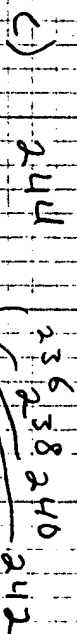
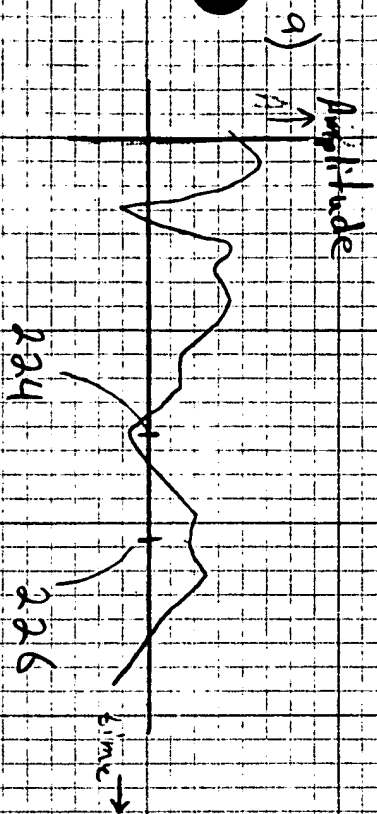


Fig. 31

09404102, 0923599

66E260" 20T40460

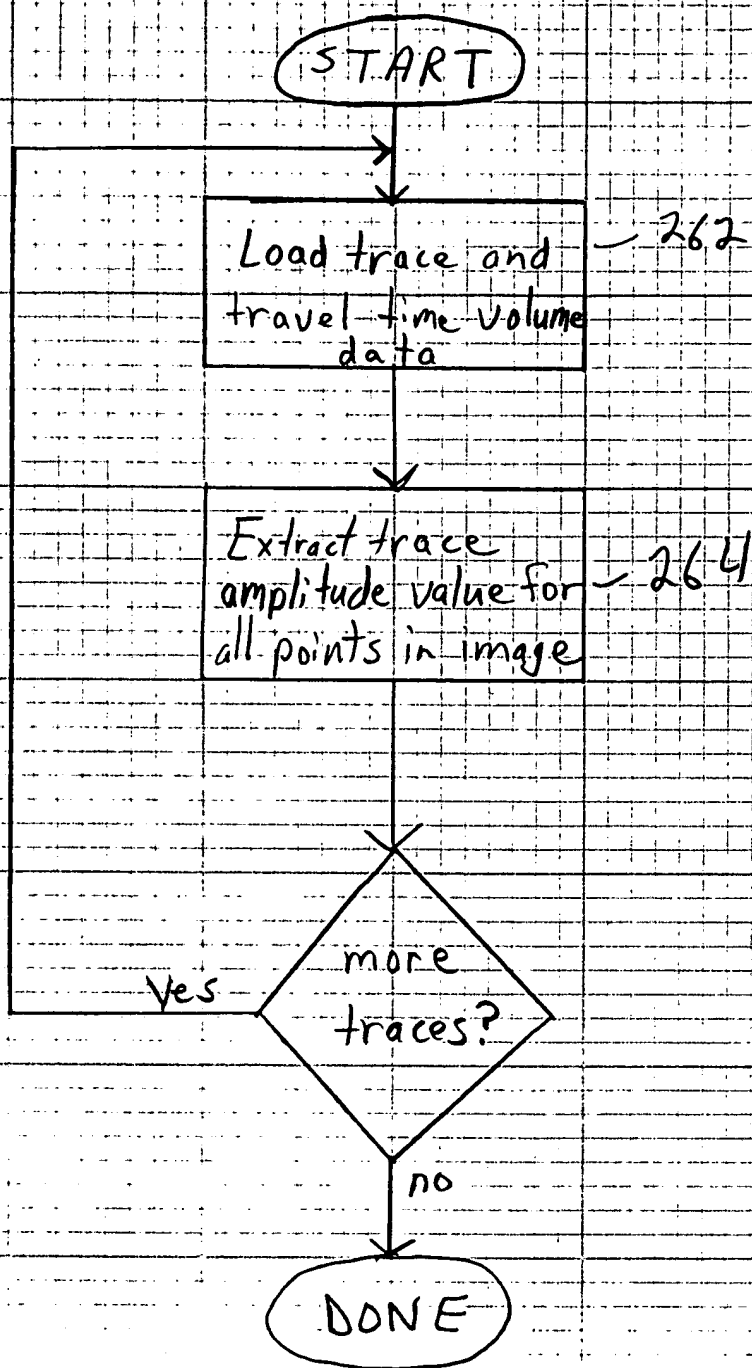


Fig. 32

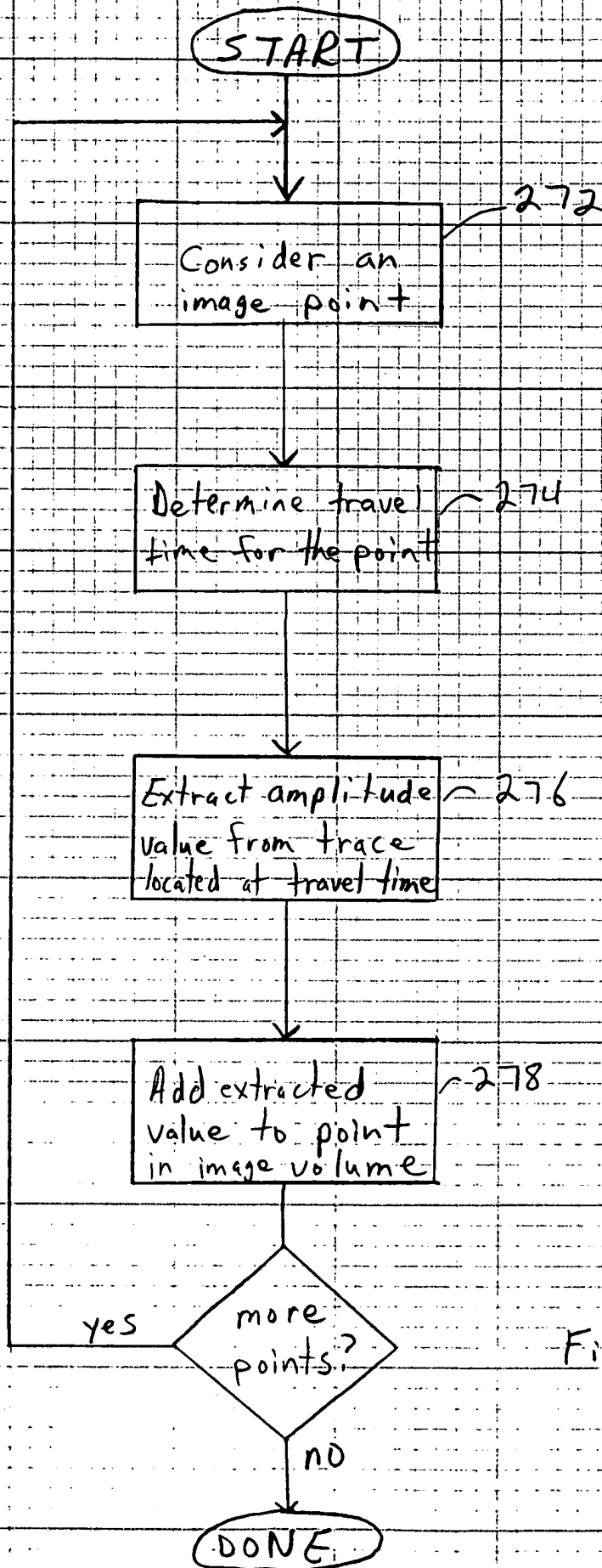


Fig. 33

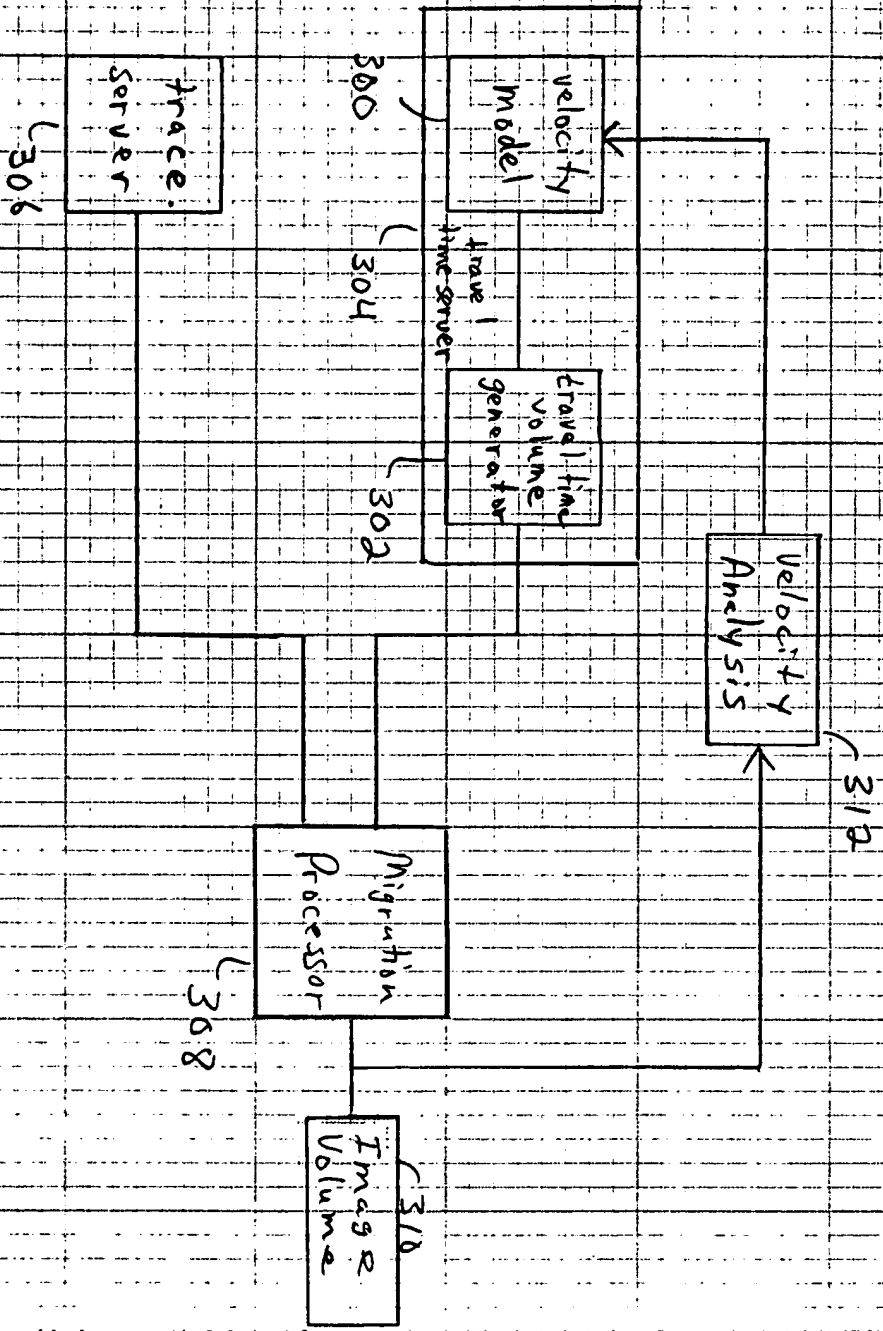


Fig. 34

09404102, 092399

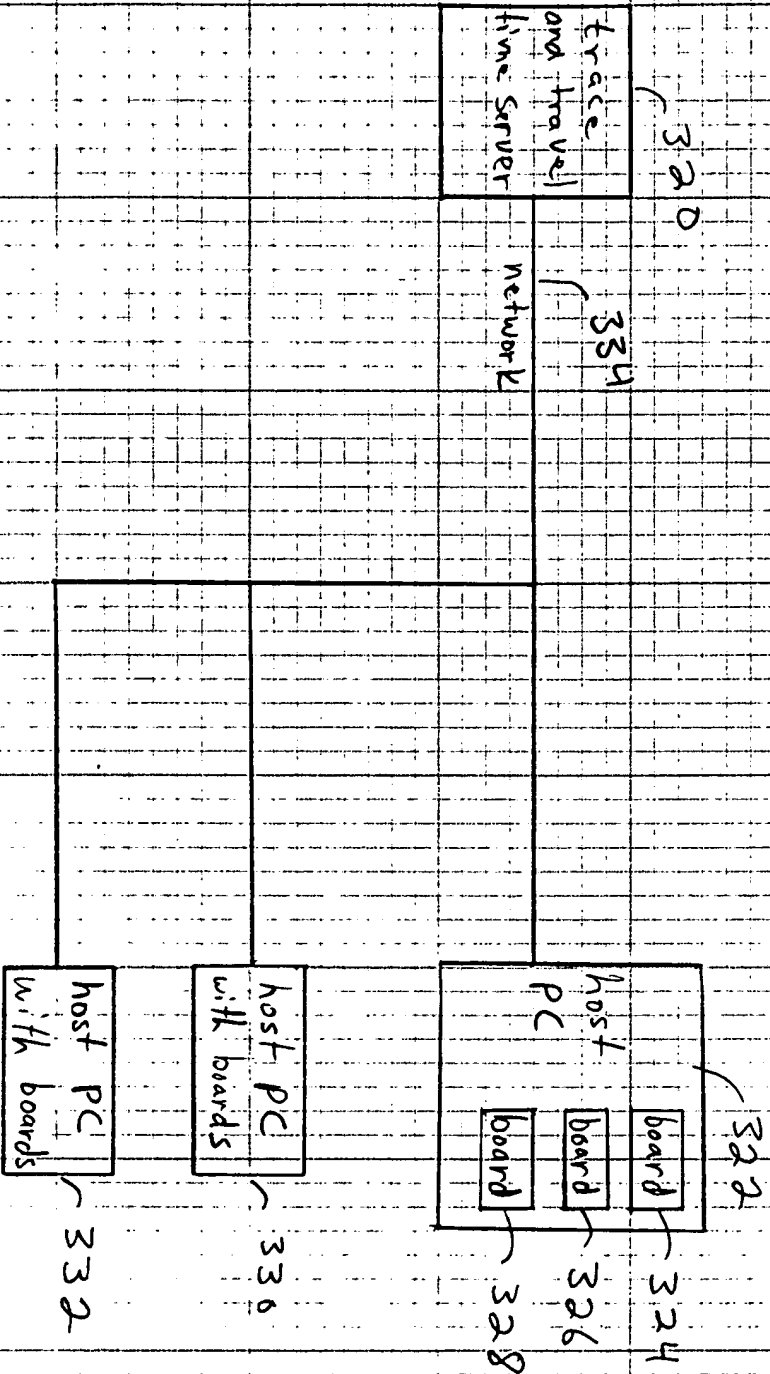


Fig. 35

Migration Processor Top Level

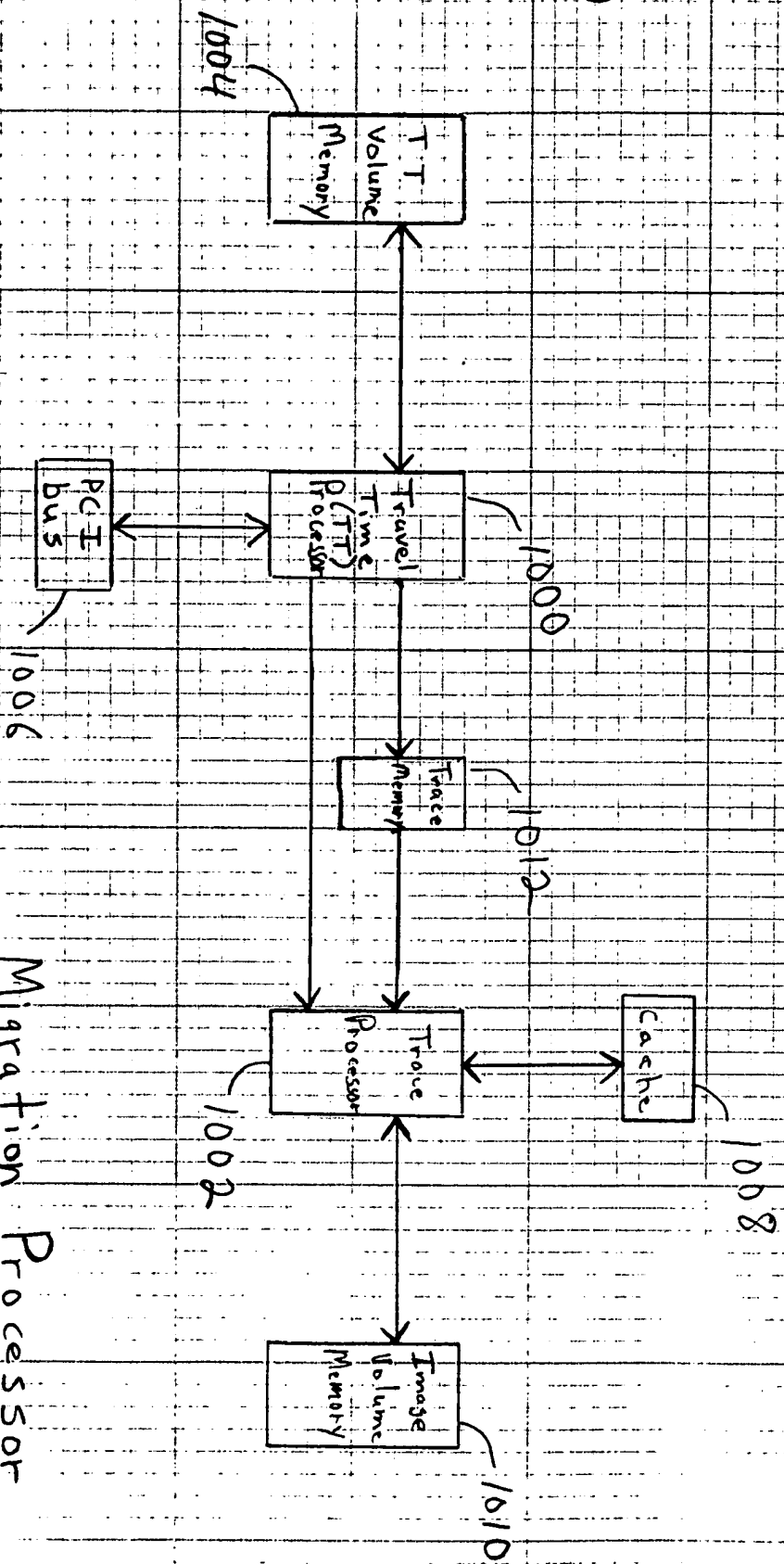


Fig. 36

09404102, 092399

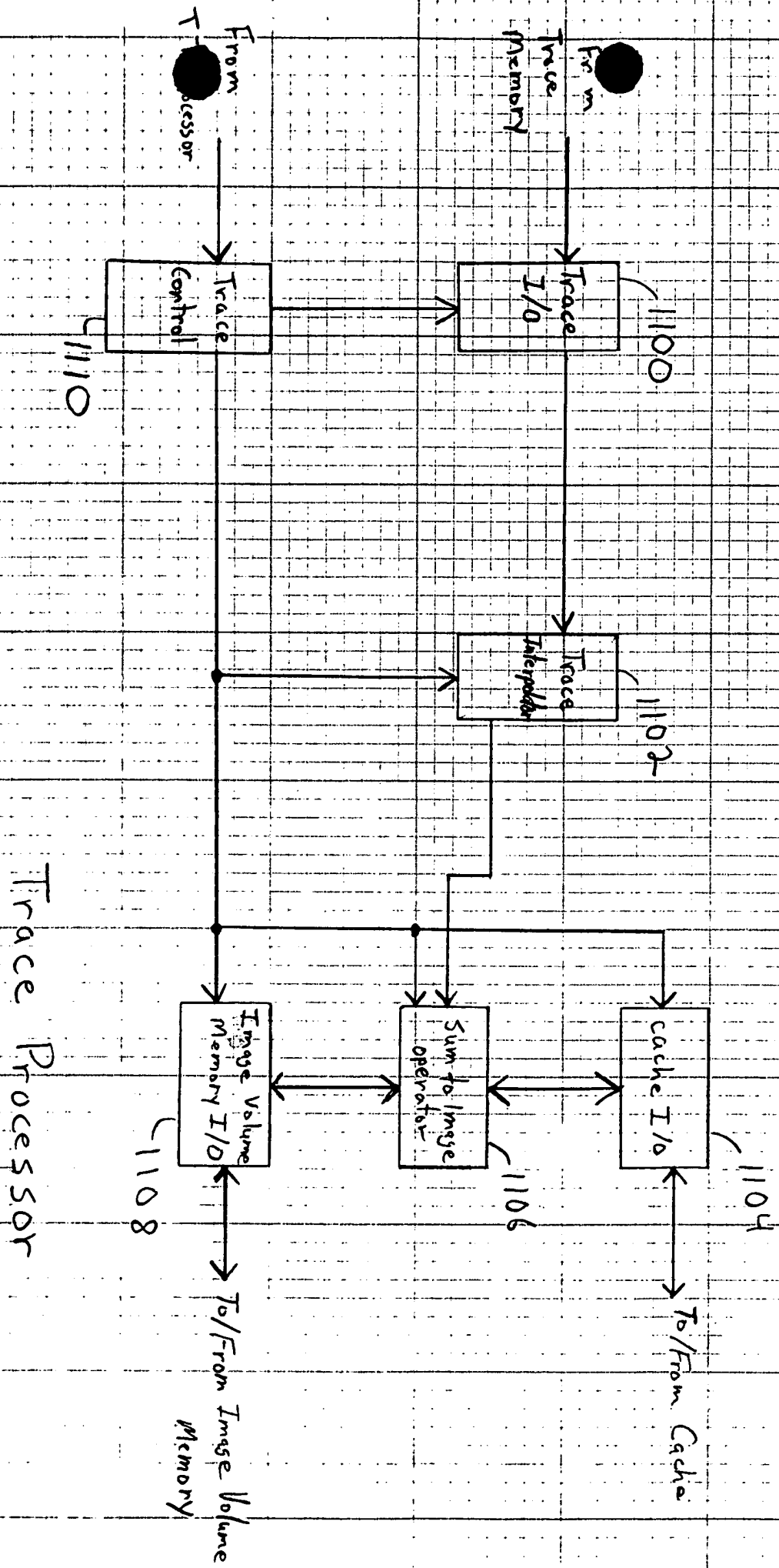


Fig. 37

Trace Processor

09404102, 092399

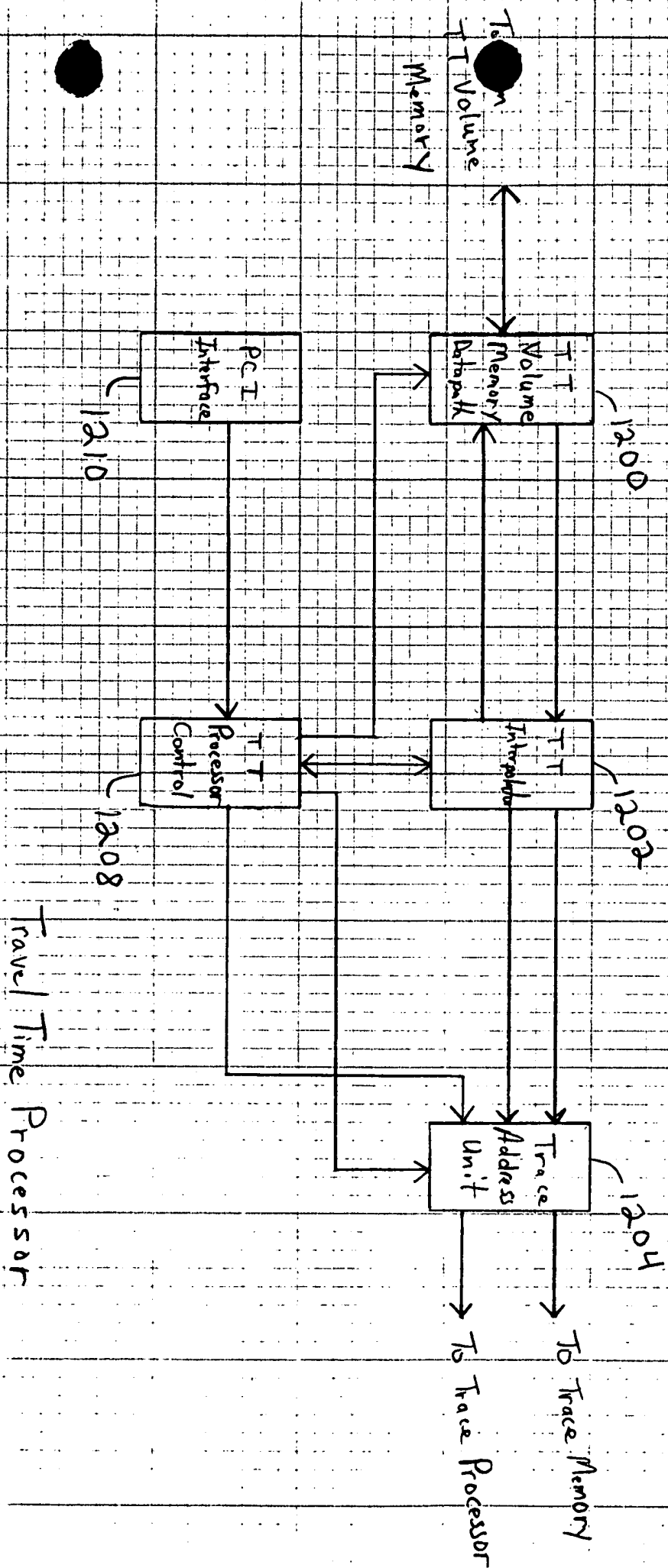
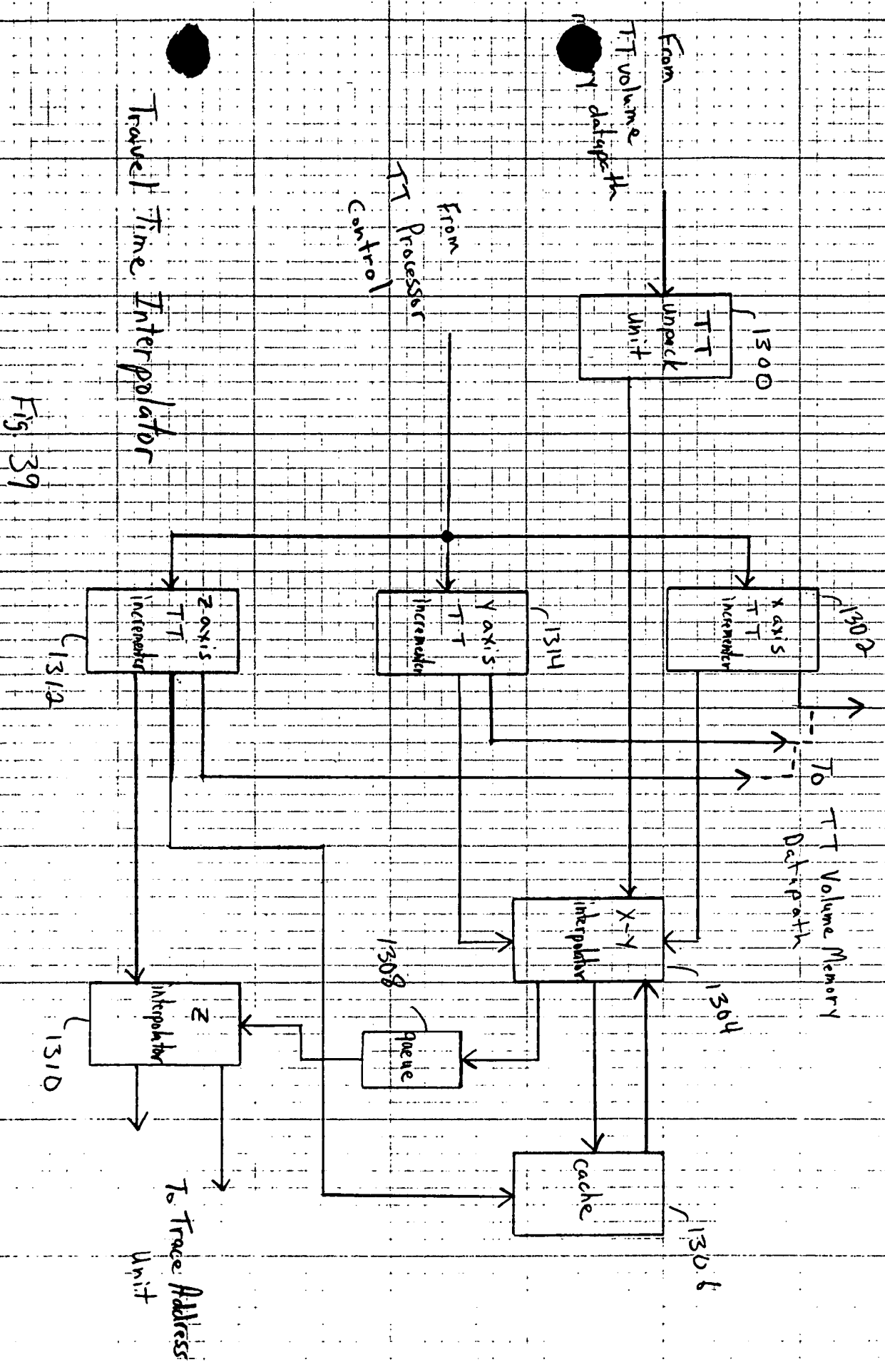
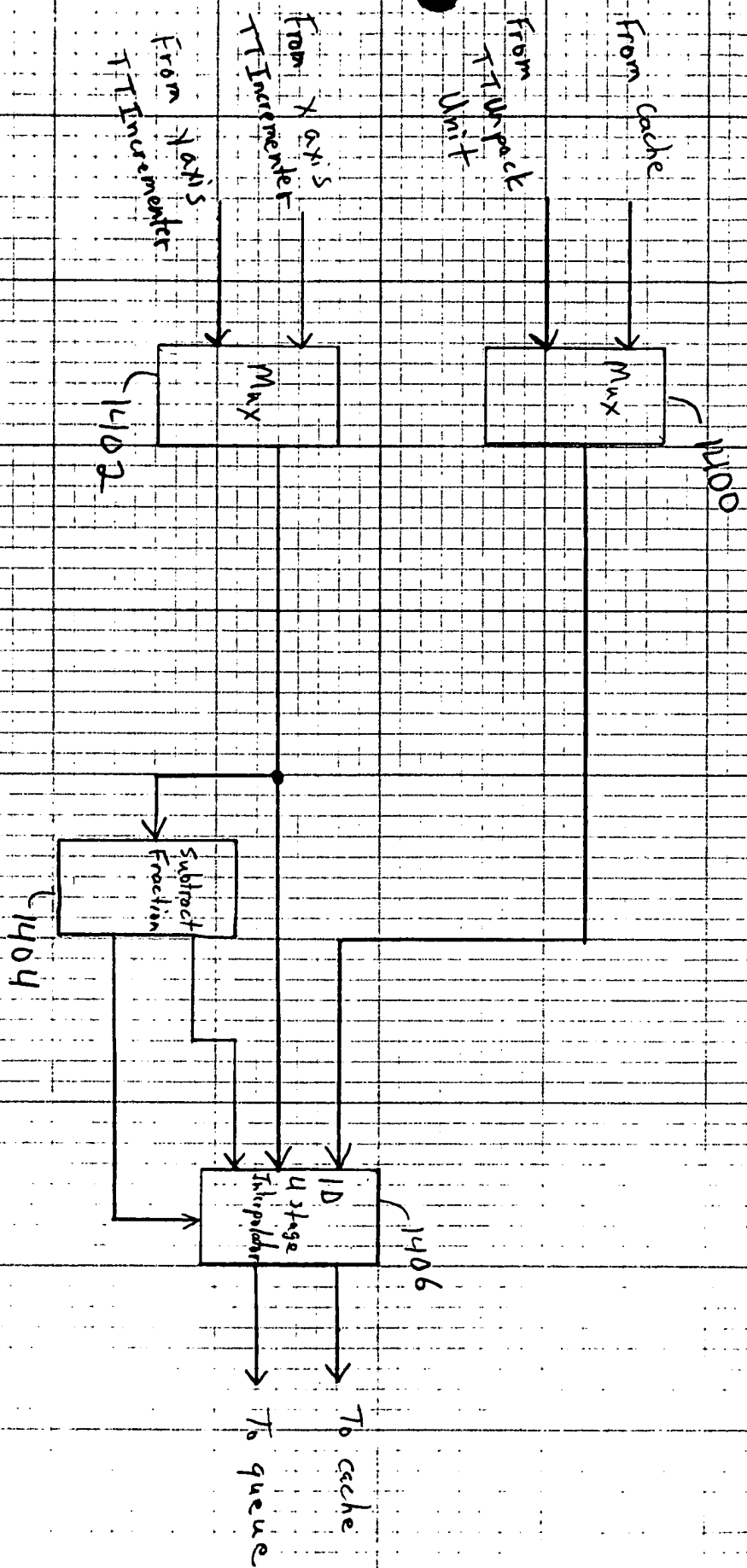


Fig. 38

09404102, 092399



09404402 092399



X-Y Interpolator

Fig. 40

09404102, 092399

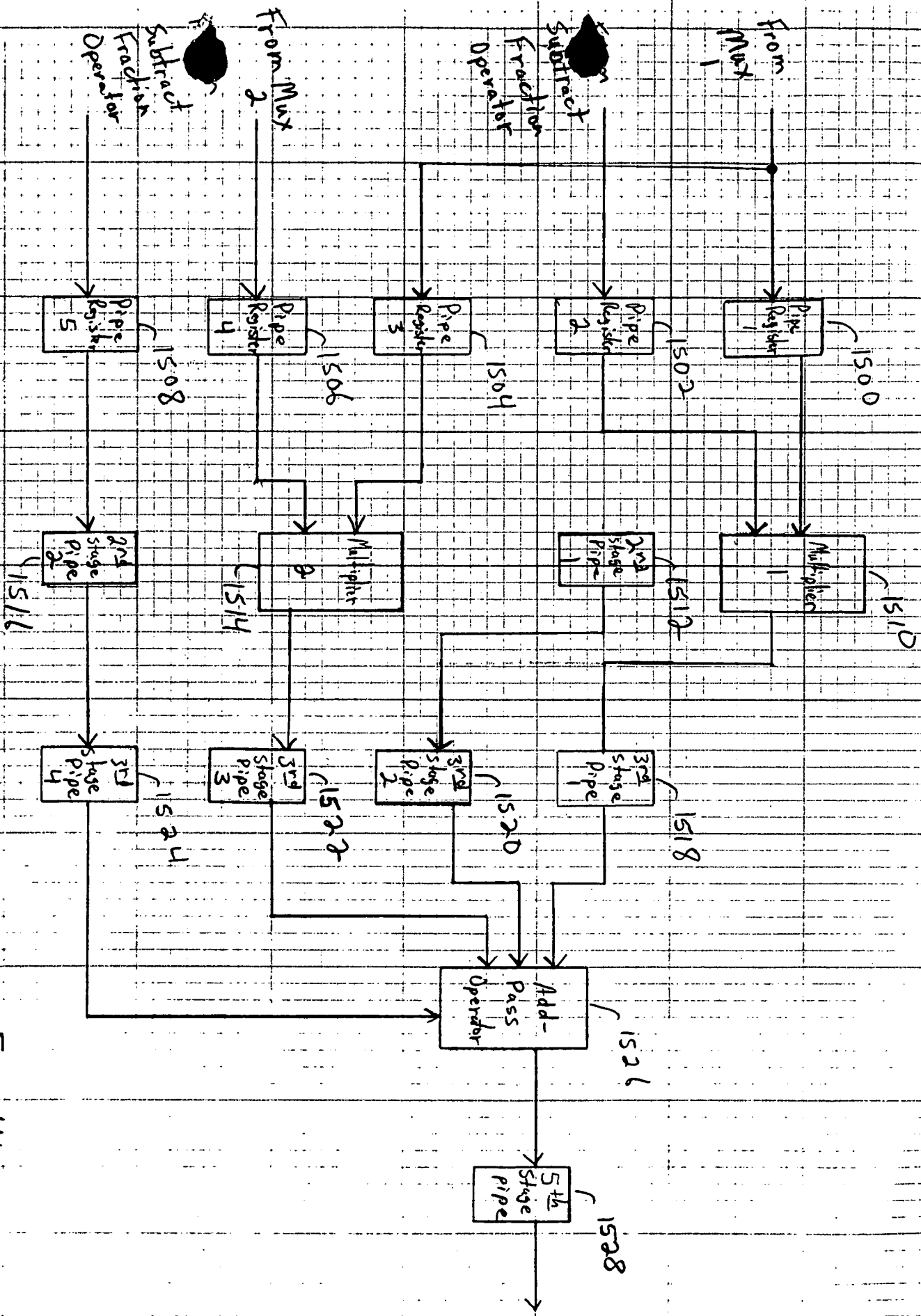


Fig. 41.

09404102 092399